FontLAB

BitFonter 3

EDITS, CREATES, TRANSFORMS, CONVERTS –
ANY BITMAP FONT IN ANY FORMAT

USER'S MANUAL FOR WINDOWS

BitFonter 3

for Windows®

User Manual

BitFonter 3 user manual, edition 3.0 [28.05.2007]

Copyright © 1992–2007 by Fontlab Ltd. All rights reserved.

Editors: Sasha Petrov, Adam Twardoch, Ted Harrison, Yuri Yarmola

Cover illustration: Paweł Jońca, pejot.com

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the publisher. Any software referred to herein is furnished under license and may only be used or copied in accordance with the terms of such license.

AsiaFont Studio, BitFonter, CompoCompiler, FONmaker, FogLamp, FontFlasher, FontLab, ScanFont, SigMaker, TransType, TypeTool, FontAudit, VectorPaint and the *FontLab* logo are either registered trademarks or trademarks of Fontlab Ltd. in the United States and/or other countries.

Apple, the *Apple Logo*, *Mac*, *Mac OS*, *Macintosh* and *TrueType* are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Adobe, PostScript, Photoshop, Type Manager, Illustrator, Macromedia, Fontographer, Flash and *Freehand* are trademarks of Adobe Systems Incorporated, which may be registered in certain jurisdictions.

OpenType, Windows, Windows 95, Windows 98, Windows XP and *Windows NT* are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

IBM is a registered trademark of International Business Machines Corporation.

Other brand or product names are the trademarks or registered trademarks of their respective holders.

THIS PUBLICATION AND THE INFORMATION HEREIN IS FURNISHED AS IS, IS SUBJECT TO CHANGE WITHOUT NOTICE, AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY FONTLAB LTD.

FONTLAB LTD. ASSUMES NO RESPONSIBILITY OR LIABILITY FOR ANY ERRORS OR INACCURACIES, MAKES NO WARRANTY OF ANY KIND (EXPRESS, IMPLIED OR STATUTORY) WITH RESPECT TO THIS PUBLICATION, AND EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES OF MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSES AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

This document was created by ***Fontlab Ltd*** (http://www.fontlab.com/).

Contents

CONTENTS 3

INTRODUCTION 9

About this Manual 10

System Requirements 12

Installation Notes 12

BITFONTER USER INTERFACE 13

Basic Terms 14
- Character 14
- Glyph 15
- Font 15
- Encoding 16
- Font Family 17
- Glyph name 17
- Menu 18
- Folders and Paths 19
- Mouse 21
- Context Menu 21
- Some More Definitions 22

Getting Started 26
- Opening fonts 27
- Editing fonts 28

Customizing BitFonter's User Interface 31
- Customizing Toolbars 32
- Customizing Menus 35
- Customizing Individual Items 36
- Converting a Menu to a Toolbar 38
- Customization of the Keyboard 40
- Links to External Programs 42
- Faster Method to Customize Commands 43

BitFonter Options 44
- General Options 47
- Font Window Options 48
- Glyph Window Options 50

Bitmap Pasting Options 53
Import Options 55
Export Options 56
Image To Font Conversion 61
Outline Font Editor 62

FONT PROJECT 65

The BitFonter Standard Toolbar 66

Creating a New Project 67

Opening a Font Project 69

Working with Fonts in the Project Window 70
The Project Window Toolbar 71
Font entries in the project and their icons 71
Browsing the project font set 72
The Project Window context menu 72
Creating new font families 73
Creating new fonts and font references 73
Duplicating, copying, and pasting fonts 74
Deleting fonts and font families 74
Viewing and editing family and font properties 75

Font Project Info 76

Verifying a Font Project 79

Reverting a Font Project 80

Saving a Font Project 81
Saving BitFonter's Project File 82
Saving a Windows Font Set 83
Saving a Windows OpenType SBIT Font 84
Saving a Palm OS Font 85
Saving in PhotoFont Format 86

EDITING FONTS 87

Editing Family Properties 88

Editing Style Properties 89

Editing Font Properties 90
Basic Set of Font Names 92
Metrics and Dimensions 97
Encoding and Unicode 103
Format Specific Data 107
Extra Properties and Font Note Pages 112
File Location and Links 112
Black/White Conversion 113

Editing a Font Character Set 114
Font Window 115

Font Editing Modes 120
Using the Font Window 126
Customizing the Font Window 128
Selecting Glyphs 129
Searching for Glyphs 131
Creating New Glyphs 132
Moving Glyphs 133
Copying and Pasting Glyphs 134
Appending Glyphs to the Font 135
Deleting Glyphs 136
Transforming Glyphs 136
Renaming and Reencoding Glyphs 137
Reencoding Glyphs Automatically 140

Reverting a Font 145

Adding a Font to the Project 146

Removing a Font from the Project 147

Duplicating a Font 148

Changing Font Size and/or Resolution 149

Changing Font Color Mode 151

Blending Fonts 153
Fonts to Blend 154
Blend Options 155
Blend Example 157

Using the Window Menu 159

CREATING AND SAVING FONTS 161

Creating a New Font 162

Creating a Font from an Image 165
Preparing Images 166
Working with an Image 172
Splitting an Image 180
Creating a Font 196

Importing a Font 198
Importing from an Installed Font 199
Importing from Outlines 201

Saving a Font 203
Saving in BitFonter Binary Font Format 205
Saving in BDF Format 206
Saving in HP Soft Font Format 207
Saving in Windows Font Format 207
Saving in AFP Font Resource 208
Saving in Palm OS Font Format 208
Saving in PhotoFont Format 209

Saving as a FontLab Template 209
Saving Color Fonts 210

Exporting a Font 212
Outline Font Editor 213
Exporting Outlines 214
Exporting to Outline Pixel Font 217
Exporting to Image File 222
Exporting as a Set of Images 224

EDITING GLYPHS 225

The Glyph Window 226

Glyph Structure 229
Units of Measurement 230
Metrics 231
Outline Layer 233

Editing Glyph Properties 234

Changing the View in the Glyph Window 236
View Menu 238
Info Panel 240
Context Menu 242

Tools 243
Selecting 244
Operations on a Selection 249
Drawing 252
Other Tools 266

Bitmap Pasting 269

Operations 270
Flip Horizontal and Flip Vertical 272
Scale 273
Rotate 275
Rotate 90°CW, Rotate 180° and Rotate 90°CCW 276
Slant 277
Make Bold 278
Shift 279
Clear Bitmap 279
Brightness/Contrast 280
Hue/Saturation 281
Threshold 282
Invert 284
Colorize 285
Remove Background 288
Emboss 289
Smart Blur 290
Gaussian Blur 291
Enhance 292
Reduce Noise 293

Median Noise 294
Sharpen 295
Soften 296
Custom Filter 297
Find Edges 299
Outline Operations 300
Example of Operations Usage 305

EDITING METRICS 307

Editing Metrics in the Glyph Window 308

Metrics Window 309
Editing Modes 310
Metrics Ruler 311
Metrics Table 312
Context Menu 313
Selecting a String for Previewing or Editing 314
Selecting Preview Size 317
Previewing Metrics Lines 317
Activating and Browsing Glyphs 318
Editing Metrics 319
Editing Kerning 328
Opening Metrics Files 333
Saving Metrics Files 335

INDEX 337

Introduction

BitFonter is a powerful tool for creating and editing bitmap fonts of various formats. With BitFonter you can make any font that you want. And that's not all. Have you ever dreamed of a real color font? BitFonter will allow you to design grayscale and even full color fonts or to color your favorite black & white fonts. This opens a whole new world of possibilities! Color fonts can be used in graphic and web design as well as for other purposes. You can draw them right in BitFonter or in your favorite drawing application such as Painter or Photoshop and then paste them into BitFonter easily for final formatting.

The key features of BitFonter are:

- High-quality, easy to use bitmap editor with more than 15 tools and 20 filters
- Free-hand drawing tools with digitizing tablet support
- Grayscale and color font support
- 100 or more levels of undo
- Support of almost all bitmap font formats
- Automatic detection of glyphs in image for import
- Export of glyphs or text to image format
- Rasterization of outline fonts and autotracing of bitmap fonts
- Glyph outline preview, import and export
- Outline pixel fonts export
- Export of outlines to FontLab vfb format
- Easy-to-use drag-and-drop interface
- Context menus and property panels everywhere
- Easy glyph metrics editing
- Support for kerning in photofonts

About this Manual

This manual covers the Windows version of BitFonter 3.

The following chapters describe all of BitFonter's features in full detail. They are organized to cover all the functions in their usual sequence.

If you are not well acquainted with fonts and font editing tools we recommend you start reading this manual beginning with the ***BitFonter User Interface*** (on page 13) chapter. It briefly covers the basic function of the program – editing bitmap fonts.

BitFonter User Interface

This chapter explains the difference between outline and bitmap fonts and illustrates the most basic tasks you can do with BitFonter. It also covers the basic definitions of the BitFonter user interface and its customization and gives a short description of all the editing windows and panels. All BitFonter options are discussed here as well.

Font Project

This chapter introduces one of the key concepts of BitFonter: the font project. The Font Project window will help you to manage the fonts you are working with.

Editing Fonts

Here you will learn how to edit a new or existing font; how to change its parameters; and how to prepare it for saving in the appropriate format.

Creating and Saving Fonts

This chapter describes the procedure of new font creation and export in full detail. You will learn how to import fonts from different sources as well as how to export new or edited fonts to an image format or outline font format.

You can draw glyphs in a drawing application or right in BitFonter or scan them from your desktop scanner and then create screen or printer fonts using BitFonter's autosplitting algorithms.

Editing Glyphs

This chapter will help you edit character shapes. Everything you can do to the glyph's image is described here.

Editing Metrics

This chapter describes the Metrics window interface, glyph metrics and kerning editing.

System Requirements

The Windows version of BitFonter requires one of the following hardware and software configurations:

A PC computer capable of running one of the following versions of Windows: Windows 98, Windows ME, Windows NT 4.0, Windows 2000 or Windows XP with one of these operating systems installed.

At least 10Mb of free space on the hard disk drive and at least 64 MB RAM. BitFonter will start on 32 MB RAM but you will need more RAM to open bigger fonts.

Installation Notes

BitFonter 3 DEMO is fully functional when opening, importing and working with fonts and images but has the following restrictions:

1. When saving fonts in any format it distorts 50% of glyphs.
2. When exporting outlines to vfb format or outline font editing application it distorts 50% of glyphs.

BitFonter User Interface

Before talking about fonts and the BitFonter font-editing features, let's spend some time learning the BitFonter user interface. For the most part, it is similar to the user interface of your operating system. If you know how to navigate in Windows and in Microsoft Office, you will feel comfortable with BitFonter. Some other parts of the user interface are unique — that is what we will focus on.

The user interface of BitFonter is highly customizable. This chapter discusses how you can custom-tailor the BitFonter interface so it fits your needs best. Note that throughout the book, we will refer to menu commands, toolbar buttons and keyboard shortcuts as they appear in the default user interface settings of BitFonter, so keep this in mind if you have customized their location or appearance.

Basic Terms

First let us define some terms that are critical to understanding BitFonter and fonts in general.

Character

The minimal, atomic unit of writing with a clearly defined identity — a part of the alphabet, a letter, a digit, an ideogram, a symbol.

Any image that can be recognized as having the same meaning represents the same character:

All the images above represent the character “A”.

Please note that sometimes, identical images represent different characters:

A Latin “A”

A Cyrillic “A”

A Greek “Alpha”

Characters are abstract beings without a particular, strictly defined image. Computers store characters in their memory using numerical *codes*. A text file contains sequences of such codes that represent strings of characters.

Glyph

The basic, atomic element of a font, the particular image that is being shown on screen or printed. The glyph repertoire of a font is a collection of all glyphs contained in this particular font. Typically, one glyph is a graphical representation of one character. However, the same font can include several glyphs that are different graphical representations of one and the same character:

Also, one glyph can represent several characters, for example in a ligature.

Characters are the abstract, conceptual components of a text, while glyphs are the particular, visual components of a text fixed in some form.

In addition to the visual appearance (the glyph image), a glyph also has some digital representation. A glyph can be represented by a bitmap, capable of reproducing the glyph image only in one specific size. More commonly, a glyph consists of outlines that are scalable so that they can reproduce the glyph image in any size.

Font

A typeface is a particular artwork of an alphabet, a set of glyphs that are designed with a common graphic idea in mind.

A font is a digital file (or a few files) that holds a digital representation of a typeface. A font contains an organized collection of glyphs along with some additional information that defines the spatial relations between the glyphs (metrics and kerning) as well as some central parameters such as names, copyright information, linespacing values etc. (font header).

Encoding

When the user presses a key (or a combination of keys) on a keyboard, a numerical code is stored in the computer's memory. This code represents a character, an abstract unit of writing. A series of such codes forms a string of text.

Every keystroke stands for a different character, so every character uses a different numerical code (also called codepoint). Operating systems and applications need to know which number represents what character — otherwise a spelling checker couldn't recognize words that you're typing. So each operating system and each application uses a list that maps characters to numerical codes. This mapping is called "text encoding". In the past, different computers used different text encoding standards (so-called codepages), so that the letter "Ä" was stored as number 142 in one program and as number 128 in another program. Modern operating systems and applications encode text using the international Unicode Standard, which assigns a unique numerical codepoint to every character used in writing by humankind. In Unicode, "Ä" always uses the code U+00C4 (which is hexadecimal for 196).

Whenever the text is printed or displayed on the screen, the computer looks inside the font file and finds out which character codes correspond to which glyphs, so that the series of abstract character codes can be visualized using some specific images of letters, digits and other symbols. Every font includes a mapping of character codes to glyphs — this mapping is called the "font encoding" (sometimes, "encoding vector", or, in this manual, just "encoding"). These days, most font formats use the Unicode Standard as basis for their font encoding, but it is possible to produce fonts that are encoded using the older codepages. It is even possible to have several different encoding vectors in one font, so that both old and modern applications can work with it.

Font Family

A font family is a collection of fonts representing typeface designs that share the same design idea but differ in width, inclination (upright or italic), stroke thickness (weight), stroke endings (serif or sanserif) or in some other stylistic aspects.

For example, “Times Bold Italic” is a typeface that belongs to the “Times” family. “timesbi.ttf”, “Times-BolIta.otf” and “tmbi_____.pfb” are all different fonts representing the same typeface in different formats.

A font family usually contains from one to a few dozen typefaces.

Glyph name

Each glyph in a font contains the digital representation of the glyph image (in form of outlines or bitmaps). In addition, each glyph is uniquely identified by its name. Glyph names must follow certain conventions:

1. Only plain English letters (uppercase A-Z or lowercase a-z), European digits (0-9) as well as the special characters “.” (period) and “_” (underscore) are permitted in glyph names.
2. Spaces are not permitted in glyph names!
3. Glyph names must not start with a digit.
4. Glyph names must not start with a period except the glyph name “.notdef”. “.notdef” is a special glyph that is displayed by the operating system if the font does not include a codepoint that the application is attempting to display. Usually, “.notdef” has a form of a rectangle, a crossed rectangle, or a rectangle with a question mark inside.

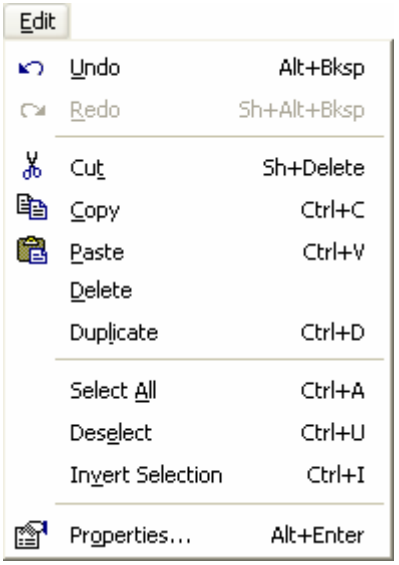
Menu

When we refer to menu items in the main BitFonter menu, we will use the following notation:

[top menu item] > [sub-item]

For example:

Edit > Copy means: click on the word Edit on the menu bar and select the **Copy** command from the menu:



Folders and Paths

Recent applications from Fontlab Ltd. use a new folder structure for storing their data files such as encoding or codepage definitions, glyph generation recipes, text samples for metrics and kerning, mapping tables etc. BitFonter 3 looks for data files in four different folders.

Shared default data folder

typically, C:\Program Files\Common Files\FontLab

This folder holds files that are commonly used by all recent Fontlab Ltd. applications: BitFonter 3, FontLab Studio, TransType SE/Pro, TypeTool, SigMaker, with more to come. In each respective subfolder, codepage definitions, encoding definitions, glyph-to-Unicode mapping files and some special data files are stored. Only Fontlab Ltd. applications and applications from registered Fontlab Ltd. developer partners should place their files there. This is to rule out conflicts between the user's customized files and default files.

Shared user data folder

typically
C:\Documents and Settings\Your Username\My Documents\FontLab\Shared

This folder has exactly the same structure as the folder discussed above and can store any files customized by the user. Any file placed in the respective location within that folder will override the corresponding file placed in the **Shared default data folder**. Please put your customized files in this folder.

Application default data folder

typically C:\Program Files\FontLab\BitFonter3

This folder holds files that are used only by BitFonter. In each respective subfolder, metrics, kerning and other text strings, additional encodings as well as samples are stored. Only Fontlab Ltd. applications and applications from registered Fontlab Ltd. developer partners should place their files there. This is to rule out conflicts between the user's customized files and default files.

Application user data folder

typically
C:\Documents and Settings\Your Username\My Documents\FontLab\BitFonter3

This folder has exactly the same structure as the folder discussed above and can store any files customized by the user. Any file placed in the respective location within that folder will override the corresponding file placed in the **Application default data folder**. Please put your customized files in this folder.

When we refer to one of the folders, we will use the following syntax:

[main folder]/[subfolder name]

Where [main folder] can be one of the following: [Shared default data folder], [Shared user data folder], [Application default data folder], [Application user data folder], and [subfolder name] is the name of the particular subfolder within that folder.

For reasons of brevity, we will sometimes write:

[Shared] which will mean either [Shared default data folder] or [Shared user data folder]

[Application] which will mean either [Application default data folder] or [Application user data folder]

This means that a particular file can be stored in either of the two locations (default or user). Remember that user locations always override default locations.

Mouse

Click the mouse on some object	Position the mouse cursor on the object and click the left mouse button
Right-click on some object	Position the cursor on the object and click the right mouse button
Ctrl-click on something	Position the cursor over “something”, hold down the **CTRL** key on the keyboard and click the left mouse button.
Drag some object	Position the cursor on the object, press and hold down the left mouse button and move the mouse to move the object, without releasing the mouse button while moving. Release the mouse button when you’re done.

Context Menu

Most windows and panels in BitFonter have context menus associated with them. These menus contain the most useful or most frequently used operations that the user may have to perform in a particular situation. To open the context menu, right-click (see the section ***Mouse*** (on page 21)) on an empty area in the window or panel, or on a particular object (e.g. a glyph or a selection or a node). Remember that the context menu will change depending on the context, i.e. right-clicking on a particular object will often display a different context menu than right-clicking on an empty area of a window.

Some More Definitions

AAT (Apple Advanced Typography) fonts
the TrueType fonts especially designed for use with ATSUI. Like OpenType fonts these fonts have special features such as swashes, contextual forms, ligatures etc. These fonts are widely presented among system fonts in Mac OS X and are supported in Cocoa applications.

Advance width
the distance between the left and right margins (sidebearings) of a glyph. Sometimes it is called *glyph width*.

AFM (ASCII Font Metrics) file
a *text* file that contains the metrics information for a PC Type 1 font.

Alphabet/Script
the collection of characters used to write a particular language. "The" alphabet (as North Americans and English know it) is the script for the English language, Latin script is the script for most European, South American and some Asian languages. Cyrillic script is used in all Slavonic languages (Russian, Ukrainian, Serbian, Bulgarian and many others). Note that a script usually includes many more characters than necessary for the one language. Latin script, for example, includes more than 200 characters.

ATSUI (Apple Type Services for Unicode Imaging)
Apple's technology and a set of routines that enable the rendering of Unicode-encoded text with advanced typographic features. It automatically handles many of the complexities inherent in text layout, including the correct rendering of text in bidirectional and vertical script systems.

BFB
the internal font format of BitFonter. It has the same function as the VFB format in ***FontLab Studio*** (http://www.fontlab.com/studio/) or ***TypeTool*** (http://www.fontlab.com/typetool/).

BMP
the standard graphics file format on Windows-compatible computers.

CMap (character map)
a table relating an encoding to a set of internal computer codes. For instance the computer may use the numbers between 1100 and 1356 to represent the characters in a font. When it needs character number 1234 it looks at the CMap table to find the corresponding code, which, in turn, directs it to the appropriate glyph.

EPS
Encapsulated PostScript. Image file format that supports both vector graphics and bitmap images.

FOND resource
Macintosh terminology for the part of a Macintosh font that contains metrics information and describes the contents of a suitcase.

FontLab format
see *VFB format*

GIF
Graphic Interchange Format. A common format for image files often used on the WWW. Resolution is limited to 256 colors.

INF (Information) file
a text file that contains information about a PC Type 1 font.

Multiple master font
a special type of font format which is an extension of the Type 1 font format. Multiple master fonts contain several font styles, called master fonts, in one font file. A program that uses multiple master fonts can not only select one of the master fonts, but it also can select an intermediate design created by the linear interpolation of the master fonts.

NFNT resource
Macintosh terminology for the part of a Type 1 or TrueType Macintosh font that contains the bitmap font.

OpenType
OpenType font format, jointly developed by Microsoft and Adobe. OpenType fonts can be TrueType-flavored (we call them OpenType TT fonts) and PostScript-flavored (OpenType PS fonts). Both are Unicode-encoded and support special features like swashes, contextual forms, ligatures etc.

PFB (Postscript Font Binary) file
a binary file that contains the glyph outline information for a PC Type 1 font.

PFM (Postscript Font Metrics) file
a *binary* file that contains the metrics information for a PC Type 1 font.

PICT
The native graphic file format for Macintosh images. A PICT file may contain black and white, grayscale, or color information.

PNG
Portable Network Graphics. A standard graphics file format designed for WWW to replace the *GIF* file format.

POST resource
Macintosh terminology for the part of a Macintosh font that contains an Adobe Type 1 font.

RGB
the color model in which red, green and blue are combined in various ways to reproduce other colors on the monitor.

sfnt resource
Macintosh terminology for the part of a Macintosh font that contains a TrueType font.

Suitcase
Macintosh terminology for a file that contains information about a font or family of fonts.

Table
a set of data defining behaviors or relationships of a font. Digital fonts contain not only the drawings of their glyphs, but also information about how those glyphs should behave. Information about the spacing on each side of a glyph (metrics), how close particular glyphs should be to each other (kerning), CMaps and many other things can be kept in tables in a font.

TIFF
Tagged Image File Format An uncompressed image file format for PC and Macintosh.

Transparency
additional information in color images allowing color pixels to be more or less transparent so the background becomes partially visible.

TrueType
a font format using quadratic b-spline mathematics to describe glyph outlines. Developed and promulgated by Microsoft and Apple Computer. We use the term Windows TrueType/OpenType TT for Windows TrueType fonts here in the Manual. Windows TrueType fonts containing embedded bitmaps are called OpenType SBIT fonts.

Type 1 (Adobe Type 1, PostScript Type 1)
a font format using cubic b-spline mathematics to describe glyph outlines. Developed and promulgated by Adobe Systems.

Unicode range
the portion of Unicode dealing with a particular language or script. E.g. the Hebrew range, the Cyrillic range, the extended Latin range. A Unicode range is not limited to 256 characters. It is usually a contiguous part of Unicode.

VFB format
the internal outline font format of Fontlab products.

Getting Started

This section will explain the most basic tasks you can do with BitFonter. If you are an advanced user and familiar with font design theory, you could probably jump ahead to the next chapter.

Generally fonts are divided into two large classes: *bitmap* and *outline*.

Bitmap fonts were originally constructed as grids of black and white dots that were copied directly to an output device: screen or printer. One dot of such a font was represented by one bit of memory. Therefore every glyph looked like a bit map. Later fonts had color characteristics and glyphs had color depth (or bit depth). But they are called bitmap fonts too.

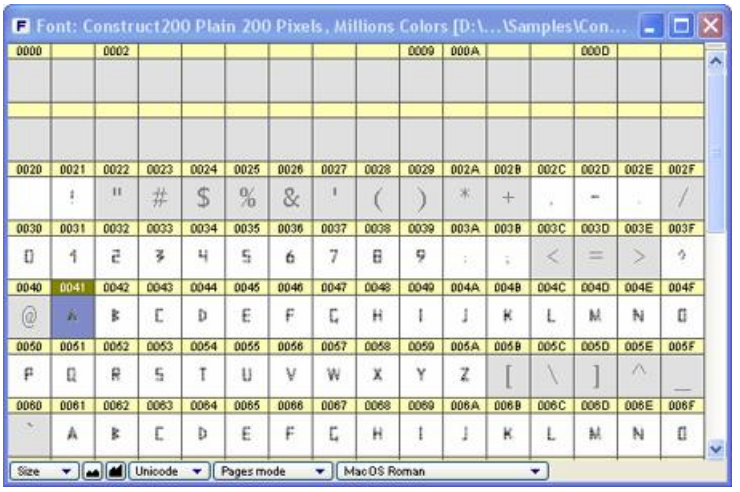
BitFonter allows you to create or edit not only black & white fonts but also color ones. Color fonts have glyphs with pixels taking more computer space than 1 bit. This includes grayscale fonts. BitFonter allows you to create 2, 4, 8, 16 and 32 bit fonts with 4, 16, 256, 32768 or 16.7 million colors respectively.

An Outline font consists of mathematically defined curves and lines and can be scaled to any size without losing its quality. PostScript (Type 1), TrueType and OpenType are outline font formats. BitFonter allows you to edit bitmap fonts only. If you are going to edit outline fonts, you must use ***Fontlab Studio*** (http://www.fontlab.com/studio/), ***Fontographer,*** (http://www.fontlab.com/fontographer/) ***TypeTool*** (http://www.fontlab.com/typetool/) or similar applications.

Let's start using BitFonter.

Opening fonts

1. Double-click on BitFonter's icon to run the program.
2. Choose the **Open** command in the **File** menu.
3. Locate the font named *Construct72* in the standard File Open dialog (it is located in the *Samples* folder in C:\Program Files\FontLab\ BitFonter3 folder, or whatever folder you installed BitFonter in) and click on **Open**.
4. The Font window named *Construct72 Plain 72 Pixels*... will open:



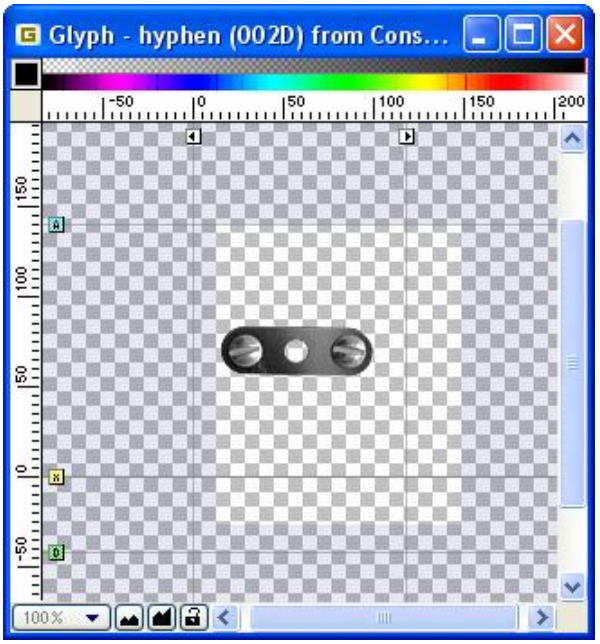
The Font window containing all the characters (glyphs) of the particular font

Here you can edit any glyph; create new ones; cut, copy or paste glyphs; view glyph information etc.

Editing fonts

5. Let's, for example, draw a 'plus' glyph. Select the **Find Glyph** command in the **Edit** menu and type 'hyphen' in the open Find Glyph window:

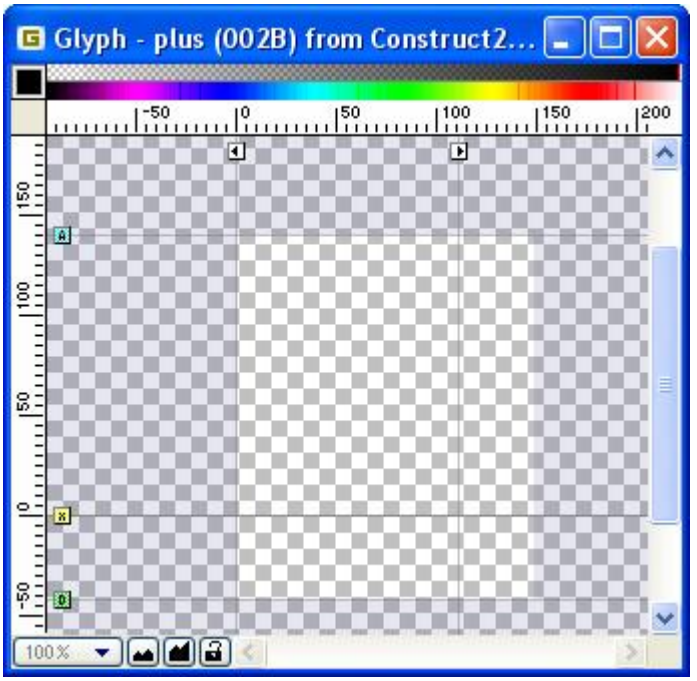
6. Click on the **OK** button and then double-click on the selected glyph in the Font window or simply press **Enter**. The Glyph window will open, allowing you to work with the 'hyphen' glyph:



Use the **Zoom In** and **Zoom Out** icons at the bottom of the window or commands from the **View** menu to enlarge or reduce the glyph view.

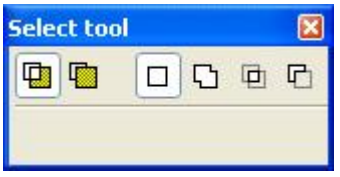
7. Choose the **Edit > Select All** command. Then **Edit > Copy**.

8. Close the Glyph window. And double-click on the 'plus' glyph in the Font window. The Glyph window will open again, allowing you to work with the 'plus' glyph. It is still empty:

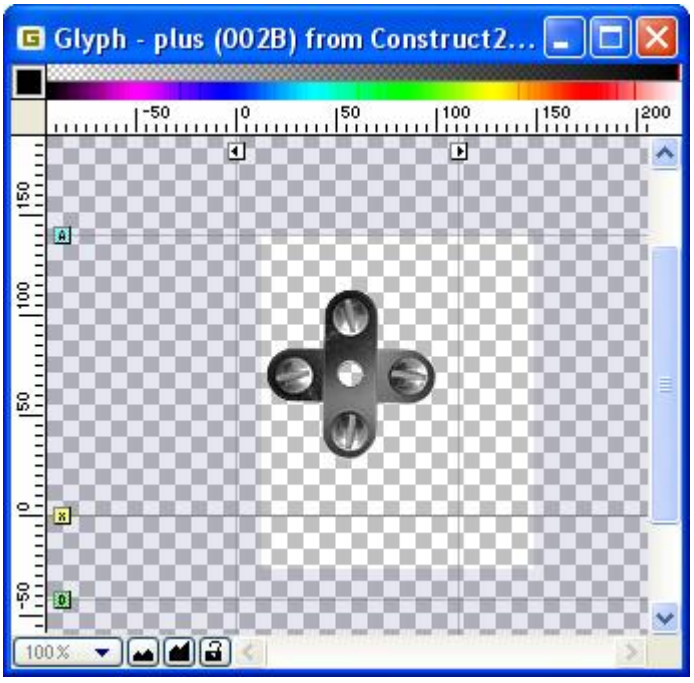


9. Choose the **Edit > Paste** command twice to paste two copies of the 'hyphen' image.

10. Choose **Tool Options Panel** in the **View** menu. Click on the **Transparent** button in the panel:



11. Choose **Tools > Rotate > 90 CW** to rotate the selection and then drag the selection to adjust its position in the Glyph window:



12. To close the font, select the **Close** command in the **File** menu or simply click on the close window button. Answer **Save** in the dialog asking about saving changes to the font.

    You have just edited the *Construct72* font.

Customizing BitFonter's User Interface

As you may infer from the title of this section most of the BitFonter user interface (which means menus, toolbars and keyboard shortcuts) is customizable. We believe our default interface is the easiest to use, but if for some reason you do not like it, you are free to make any changes you want. If you do not want to change anything in the BitFonter user interface, you can fast forward to the next section.

The general idea of customization is simple: there is a long list of commands that you can use and three kinds of controls: menus, toolbars and keyboard shortcuts. Through customization you can assign any command to a menu item, button on a toolbar or combination of keys pressed on a keyboard. In addition you can organize commands in context menus or toolbars.

Most of the customization commands are concentrated in the Customize panel that you can open with the **Customize** command from the **Tools** menu or the same command located in the context menu which appears if you right-click on a menu, toolbar or toolbar dock area:

The **Customize** dialog box consists of several pages:

Commands	List of all the available commands grouped into several categories
Toolbars	Customization of toolbars. There is an option to create new toolbars.
Tools	On this page you can "connect" an external program to a menu item in BitFonter's Tools menu
Keyboard	Customization of keyboard shortcuts
Menu	Customization of menus

While the **Customize** dialog box is open all interface elements are in "editable" mode, so you can simply drag-drop buttons and menu items between different toolbars. You can also customize the appearance of menu items and toolbar buttons.

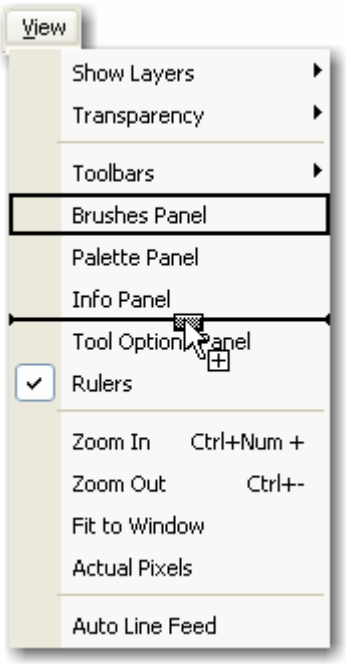
Customizing Toolbars

To move a button within a toolbar, click the left mouse button on it; drag it to the new location and drop it. If you drag the button slightly further to the right, a separator bar will be added between it and the previous button:

To move a button to another toolbar, drag-drop it there. To copy a button, hold down the **Ctrl** key while you drag the button.

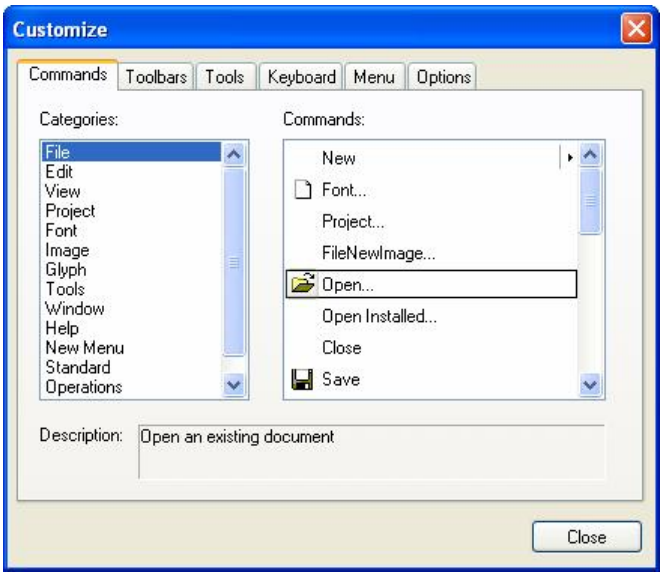
To remove a button from a toolbar, drag it out of the toolbar:

In BitFonter there is very little difference between a menu and a toolbar, so you can re-arrange, copy or remove menu items just like you did with toolbar buttons:



You can also drag a menu item onto a toolbar to add a toolbar button. Hold the **Ctrl** key to copy the item.

To get access to all the BitFonter commands, open the **Commands** page in the Customize dialog box:

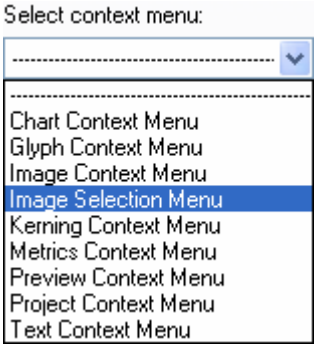


In the left list select a group of commands and use the list of commands in the right list as a source of menu items and toolbar buttons: drag the commands from there.

Customizing Menus

If you want to create a new menu, select the **New Menu** group in the left menu and drag it onto the main menu bar or any of the toolbars. A new menu appears and you can start adding commands to it using the drag-drop technique described above.

With the Customize dialog not only can you customize the main menu, but also most of the context menus which appear when you right-click on BitFonter windows. Open the **Menu** page in the Customize dialog box and choose a context menu in the right dropdown list:



A menu appears on screen and you can customize it by dragging commands from the toolbar, other menus or the list of the commands on the **Commands** page.

To reset changes you have made in menus, use the **Reset** buttons on the **Menu** page of the Customize dialog box. Use the left **Reset** command to reset changes in the main menu and the right **Reset** button to reset changes in the context menus.

Customizing Individual Items

You can customize the appearance of any menu item or toolbar button. The following appearances are available for most items:

Image

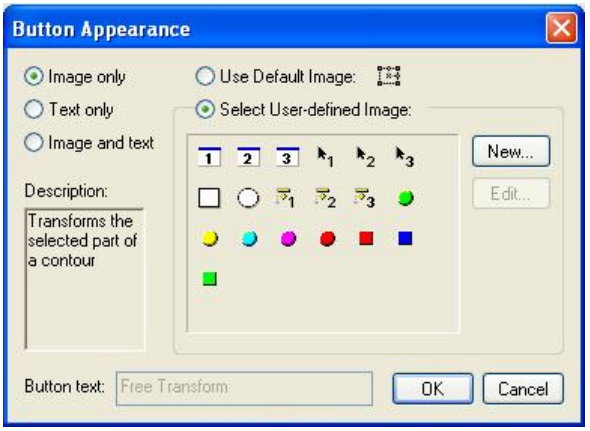
Text New Font

Image and Text New Font

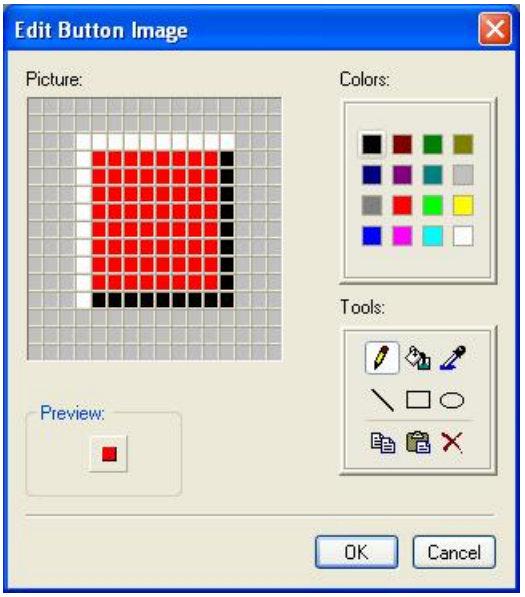
To change the appearance of the menu item or toolbar button position the mouse cursor on the button and click the right mouse button. Select the new appearance method in the context menu:

Most commands in BitFonter have pre-designed images, but you can easily create your own images for any toolbar button or menu command. To do so, select the **Button Appearance** command in the button's context menu:

You will see a dialog box where you can choose the appearance method and, if it includes an image, choose the image that appears on the button or at the left of the menu item:



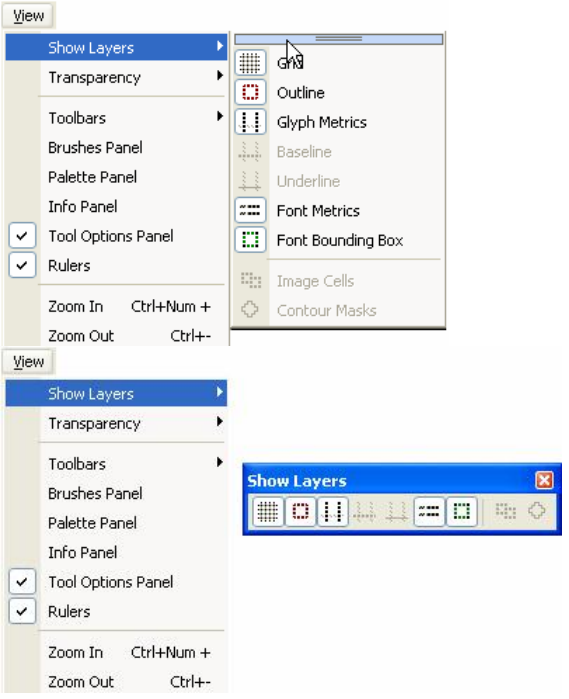
Choose a User-defined image; click on **New** to create a new image; or **Edit** to edit one of the User images. If you decide to change the image, use the included image editor to change it:



Use one of the **Tools** to edit the enlarged image and choose a color in the **Colors** area. Click on the **OK** button when you are ready.

Converting a Menu to a Toolbar

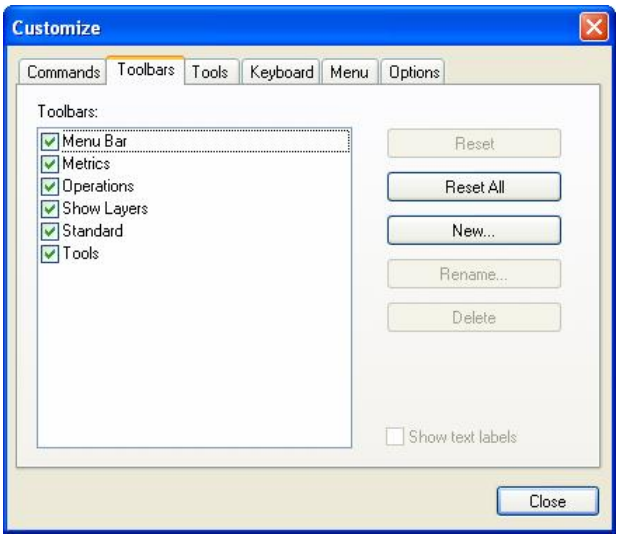
In BitFonter some menus can be converted to toolbars. If you open a menu and can see a tiny caption in the top area of it, you can drag it to any place on screen and it becomes a toolbar:



Not all menus have this feature, but you may find it really useful.

OK, that is almost all about customizing toolbars and menus. A few more things:

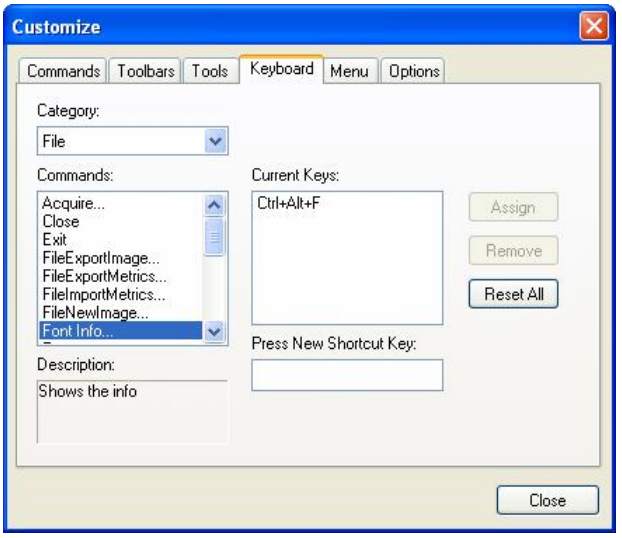
To reset changes you have made in toolbars, use the **Reset All** button on the **Toolbars** page in the Customize dialog box:



Use the **New** command on the same page to create a new toolbar. After doing this, add commands to it by the drag-drop method described earlier in this section.

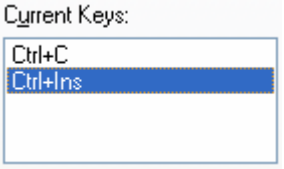
Customization of the Keyboard

Open the **Keyboard** page of the Customize dialog box:



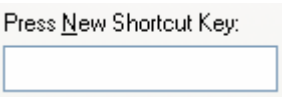
In the left area of the page you can select the command, which you want to customize. Choose the commands category in the top list and the command itself in the list below.

On the right part of the page you will see the list of keyboard shortcuts currently defined for that command:



The **Remove** command at the right of the list allows removal of one of the existing shortcuts.

To define a new keyboard shortcut, select a command and position the cursor on the editing field below the **Press New Shortcut Key** label:

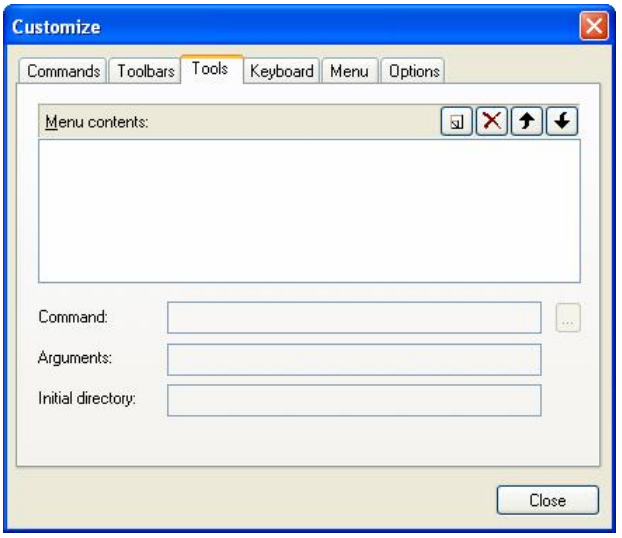


When the caret is in position, press the combination of keys that you want to assign. A description of that combination will appear in the editing field and you can click on the **Assign** button to assign that combination to the currently selected command.

Click on the **Reset All** button to reset all changes back to BitFonter defaults.

Links to External Programs

Use the **Tools** page of the Customize dialog box to assign Windows programs to menu items in BitFonter's **Tools** menu:



There is a list of the assigned programs in the middle of the page and it is empty by default. Click on this button: to add a link.

Enter the name of the program as it will appear in the menu:

Then use the button at the right of the **Command:** editing field to locate the program to run.

Editing fields at the bottom of the **Command** field allow you to define arguments for the program you want to run from BitFonter's menu.

Use the button at the top of the **Tools list** to remove the reference to the program and the and buttons to change the order of the commands.

You may use special parameters to run external programs with the currently opened font as an argument. When BitFonter recognizes this argument, it will replace it with the file name of the currently active font or with some other parameters.

Suppose that current font was last saved into file named `C:\fonts\sample.vfb`.

Special arguments are:

%p Full path of the current font [`C:\fonts\sample.vfb`]

%f Name of the file with extension [`sample.vfb`]

%n File name only [`sample`]

%x File name extension [`vfb`]

%d Folder where file was saved [`C:\fonts\`]

%a Folder of BitFonter installation [usually it is "C:\Program Files\FontLab\BitFonter3"]

Now you know everything about the customization of menus, toolbars and the keyboard, so you can click on the **Close** button at the bottom of the Customize dialog box to exit the customization mode.

➘ Important note: in the following manual we will describe all commands, buttons and keyboard shortcuts as they come with BitFonter, without any customization. If you changed the interface but want to follow the manual, reset all changes with the **Reset** buttons on the **Toolbars**, **Keyboard** and **Menu** pages of the Customize dialog box.

Faster Method to Customize Commands

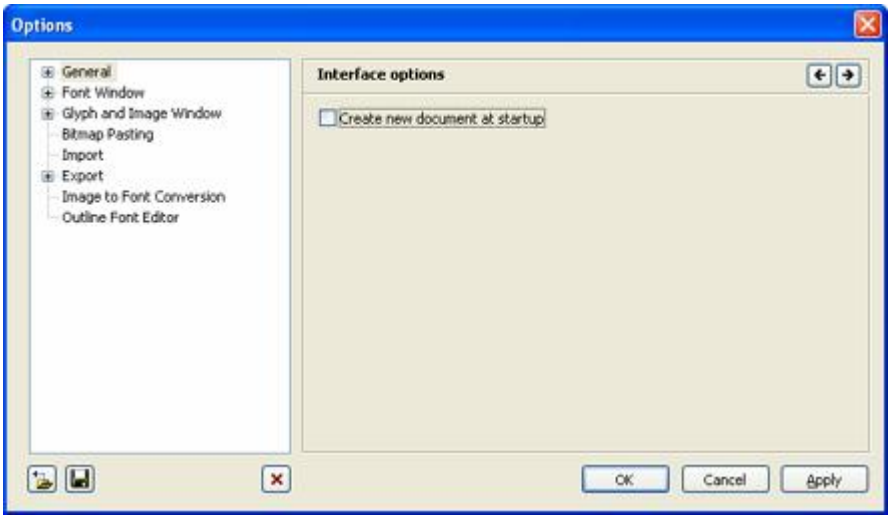
You can customize toolbars and menus without opening the Customize dialog box by holding down the **Alt** key on the keyboard and dragging buttons between toolbars or toolbars and menus.

BitFonter Options

This section is very important since it is referenced from many other sections of the Manual.

Most of the features, behavior, import and export algorithms of BitFonter are customizable in the **Options** dialog box. We encourage you to experiment with the settings and adapt them to your preferences. However, note that the authors have carefully chosen the factory settings so if you do not feel like poking around the Options, in most cases you will be fine with the defaults.

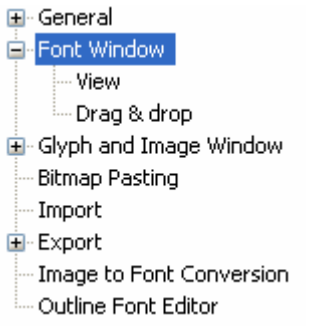
To open the Options dialog box, select the **Options** command in the **Tools** menu:



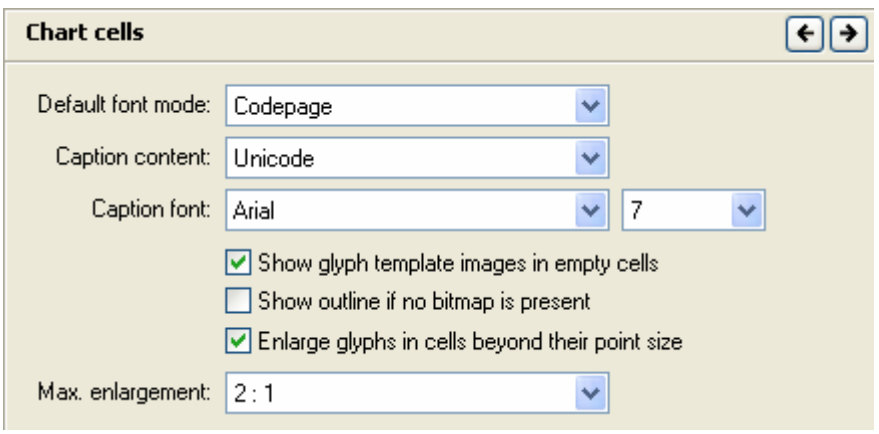
The dialog structure is quite simple. There is a list of pages combined in categories on the left, the contents of the currently selected page on the right and some buttons on the bottom. You will notice that the structure of this dialog bears resemblance to the structure of the **Font Info** dialog.

To select a page use the list on the left:

Expand one of the categories to see all the pages:



Select a page and you will see its contents appear at the right of the list:



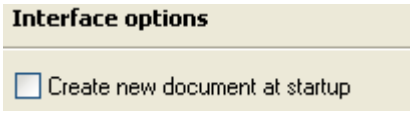
You can browse pages continuously by clicking on buttons.

Other buttons and their meaning are described in the table:

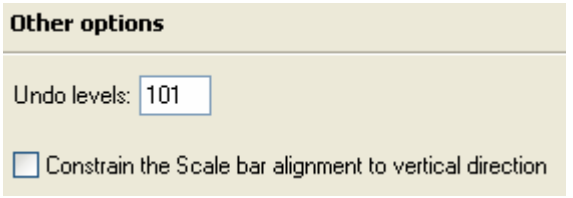
Import options	Allows you to select a profile file that holds a particular configuration of all options and loads that profile. You can create different profiles for different occasions and load them when needed — for example, separately for each format or foundry that you work with
Export options	Exports current options to a profile file. In a workgroup environment, you can export a profile file and give it to your colleague who then can load it and generate fonts in the same environment. When sending technical problem reports to Fontlab Ltd., please always export your options into a profile file and attach that file with your report
Reset options	Resets all options to the factory defaults
Apply	Applies the changes without closing the dialog box. Many interface changes become visible immediately in the corresponding windows
Cancel	Closes the dialog box without applying changes
OK	Applies the changes and closes the dialog box.

General Options

General Options pages are used to customize some interface option and the number of **Undo** levels.



To make the program open the new font creation dialog at startup, check the **Create New Document at Startup** option. The **New font** dialog will open every time the program starts.

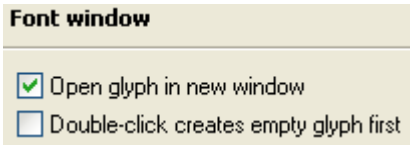


Use the **Undo levels** edit box to enter the number of undo levels. The default value is 101 undoes. Note that the more undo levels you want to use, the more memory the program will require.

Switch the **Constrain the Scale bar alignment...** option on to prevent the Scale bar from being slanted. By default this option is off and you must hold down the **SHIFT** key on the keyboard to keep the Scale bar vertically or horizontally.

Font Window Options

The next page of the Tools > Options dialog is used to customize the view in the Font window:

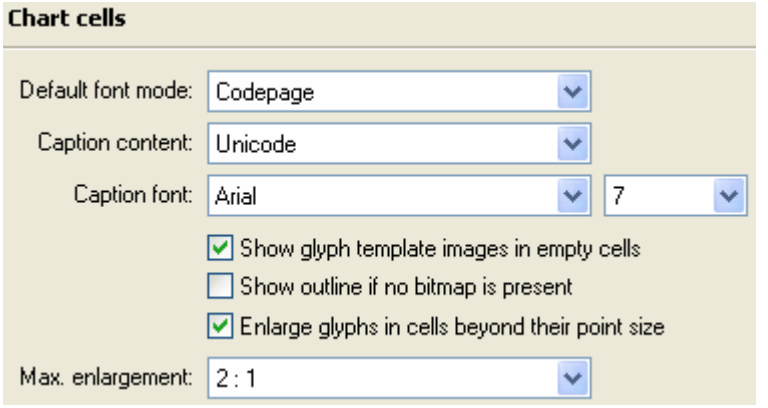


Open glyph in new window

When checked, this option tells BitFonter to open a double-clicked glyph in a new Glyph window. If this option is off, all glyphs are opened in the same Glyph window for editing. Holding down the **Ctrl** key when double-clicking on the glyph in the Font window reverts the meaning of this option.

Double-click creates empty glyph first

When this option is checked double-clicking on an empty cell in the Font Window creates the glyph but does not open it. To open the glyph for editing you need to double-click on it again. By default this option is off and double-clicking on an empty cell creates and opens the glyph.



Define the default font editing mode with the help of the **Default Font Mode** dropdown list.

Define the default cells' caption content with the **Caption content** dropdown list. Here you can choose the font and the font size for the caption.

Show glyph template images in empty cells

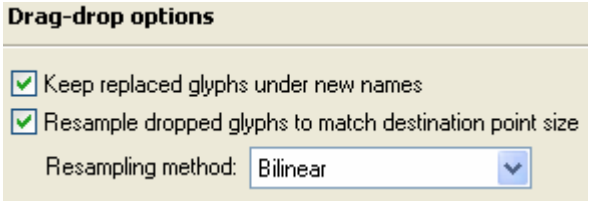
Uncheck this option if you do not want the template glyphs to be shown in empty cells of the Font window.

Show outline if no bitmap is present

Check this option if you want the cell show outlines if the glyph does not contain bitmap image. If neither outline nor bitmap is present the cell will not contain any image.

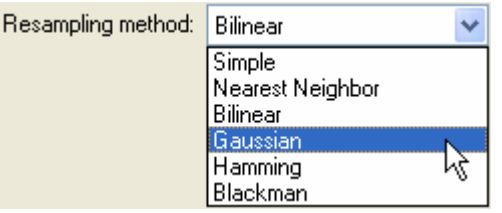
Enlarge glyphs in cells beyond their point size

This check box lets the glyph image zoom when the cell zooms. The **Max enlargement** dropdown list allows you to limit glyph zooming in its cell.



The **Keep replaced glyphs under new names** option, when checked, allows you to keep old glyphs after pasting the new ones. Replaced glyphs will get their new names.

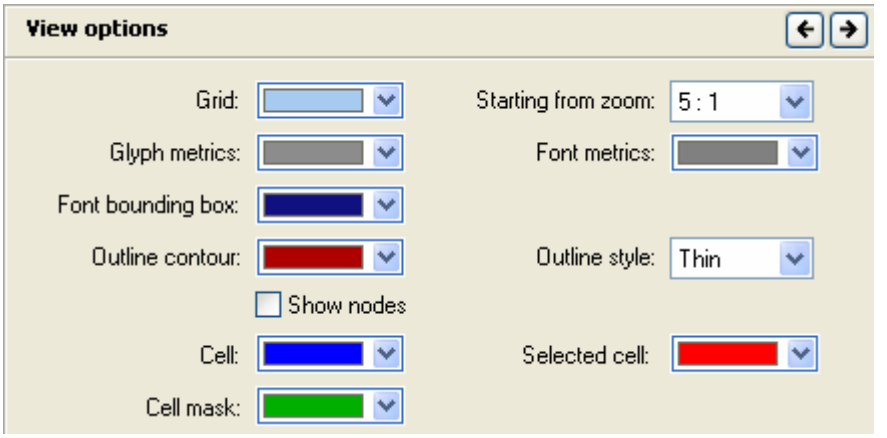
The **Resample dropped glyphs...** option, when checked, tells BitFonter to resize the dropped-in glyph images to fit the current font point size. The resampling method can be defined in the dropdown list:



Glyph Window Options

The **Glyph and Image Window** page is used to customize the view and behavior of the Glyph window as well as some elements of the Image window.

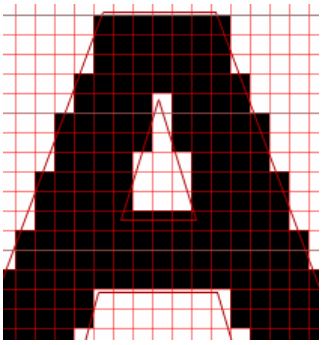
View Options



You can change colors for some elements of the Glyph and Image windows. Click on the corresponding color rectangles to change the color for grid, glyph metrics lines, font metrics lines, outlines, cells and the font bounding box rectangle.

Grid

The gridlines are not part of the transparency grid. They delimit pixels in all color modes when large zooming is selected:

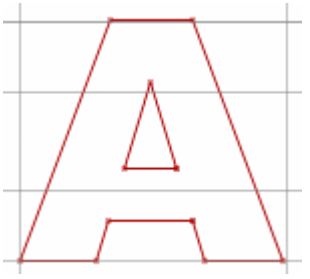


Red grid lines

Select the starting zoom mode when the grid becomes visible.

Outline contour

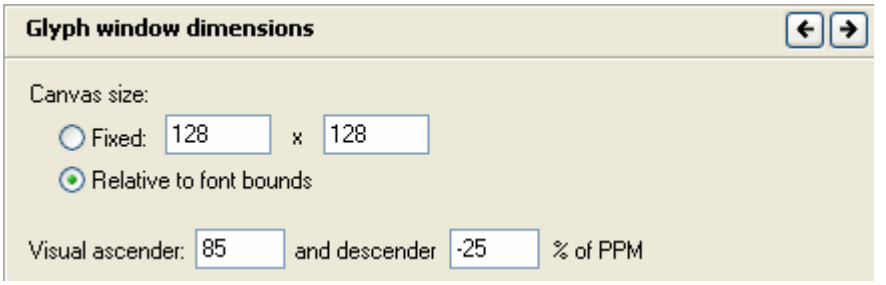
Besides the color of outline contours you can choose to **Show nodes** and/or to show outline **Thin** or **Thick:**



Thin outline with nodes shown

Thick outline without nodes shown

Glyph window dimensions



Canvas size

Fixed	The fixed canvas size means that the editable area is of custom fixed size in the Glyph window. Enter the horizontal and vertical values in the editable fields
Relative to font bounds	The relative canvas size means that the size of glyph's editable area depends on the glyph's bounding box size. This option is set by default

Visual ascender and descender

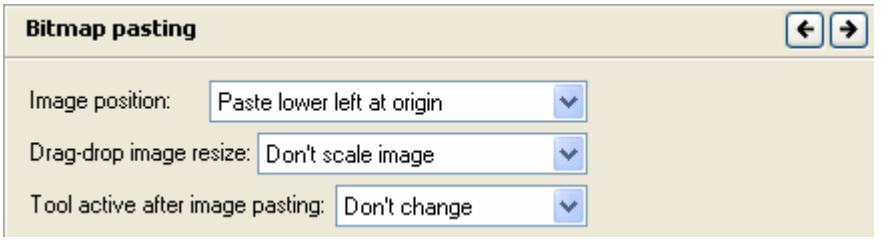
When you select 100% as the zoom value, BitFonter needs to choose a scaling factor to fit the font unit space in the Glyph window. Two vertical levels in the font space define this scaling: Visual Ascender and Visual Descender.

When you select 100% zoom, it means that Visual Ascender is fitted to the top of the editing field and Visual Descender to the bottom. The same values are used to build the icons that you see in the Font window.

Values are measured in percentage of the font PPM size, so -25% is -6 if the PPM size is 24 and -12 if the PPM size is 48.

Bitmap Pasting Options

The Bitmap Pasting Options page allows you to fine tune image pasting as done with the common **Paste** command or by drag-and-dropping:



There are three controls on the page:

Image position

This option affects the drag-and-drop pasting from the Image window to the Font window or pasting an image copied from the Image window in the Glyph window. Pasting an image from one Glyph window to another with the **Edit > Paste** command is always performed **in the same position** where copying was made.

Paste Lower Left at Origin	Choose this option if you want the lower left corner of the bitmap image to be pasted at the glyph's origin point. This option is selected by default
Paste Upper Left at Ascent	Choose this option if you want the upper edge of the bitmap image to be pasted at the font ascent line
Paste Centered in Canvas	Choose this option if you want the bitmap image to be pasted in the center of the editing field
Paste Centered in Font Bounding Box	Choose this option if you want the bitmap image to be pasted centered in the bounding box.

Drag-drop image resize

This option affects only the drag-and-drop process when you drag the selected portion of an image from the Image window to the Font window. Pasting in the Glyph window with the **Edit > Paste** command is always performed **without scaling** the image. During a drag-and-drop operation from one Font window to another glyphs are **always scaled**.

Do not Scale Image	Choose this option if you want the bitmap image to be dropped without scaling it. This option is the default choice
Reduce to fit Bounding Box	Choose this option if you want the image to be pasted fitted to the height of the bounding box. The image width will change proportionally. If the image height is less than the height of the bounding box the image will not be scaled
Reduce to fit Font Ascent	Choose this option if you want the image to be pasted to the font ascent line. The image width will change proportionally. If the image height is less than the font ascent the image will not be scaled.

If you need to copy glyphs from other applications, e.g. Photoshop, adjust the image size in Photoshop and then use the copy-paste method.

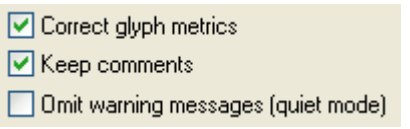
Tool active after image pasting

This option affects pasting to the Glyph window with the **Edit > Paste** command.

Do not change	If this option is selected, any pasting operation in the Glyph window will not change the currently selected tool
Select	If this option is selected, any pasting operation in the Glyph window will change the current tool to the **Select** tool
Select Region	If this option is selected, any pasting operation in the Glyph window will change the current tool to the **Select Region** tool
Edit Metrics	If this option is selected, any pasting operation in the Glyph window will change the current tool to the **Edit Metrics** tool.

Import Options

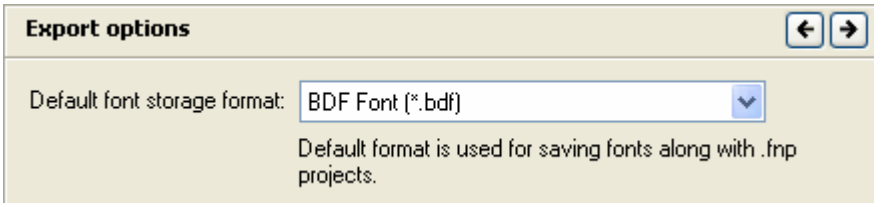
The **Import** page of the Tools > Options dialog allows you to customize font opening options:



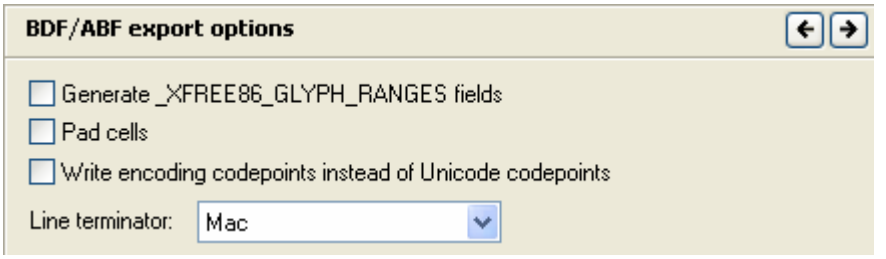
Correct Glyph Metrics	When this option is on BitFonter checks and corrects, if necessary, the glyph metrics in the font while opening it
Keep Comments	When this option is on BitFonter reads comments from the font while opening it
Omit Warning Messages	When this option is on BitFonter does not display warning messages on import when finds something wrong with fonts.

Export Options

The **Export** section of the Tools > Options dialog allows you to customize font export options:

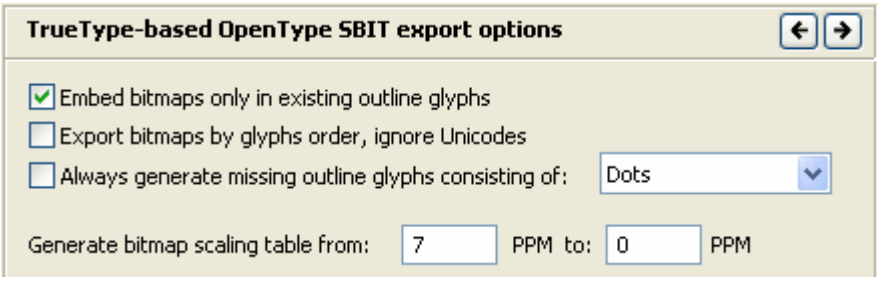

Export options
Default font storage format:
BDF Font (*.bdf)
Default format is used for saving fonts along with .fnp projects.

The **Default font storage format** dropdown list allows you to choose the font format used by default for saving fonts along with their .fnp project with the **Save a Copy As...** command or when importing multiple sizes of the installed font. Initially the default font storage format is BFB (BitFonter Document).



These options refer to saving fonts in BDF format only:

Generate XFREE86_GLYPH_RANGES fields	When this option is on BitFonter generates BDF files with special key words that define existing glyphs and their codes. Usually this option need not be switched on
Pad Cells	When this option is on BitFonter generates and saves all glyphs at the same size. Usually this option need not be switched on
Write encoding codepoints instead of Unicode codepoints	When this option is on BitFonter writes 1-byte codes for glyphs but not 2-byte Unicode codepoints. This option is off by default
Line terminator	This dropdown list lets you define the end of line style in a BDF file. You may choose among Mac, DOS and Unix line terminators. DOS EOL mark is the default value.



The next group of options is about saving fonts in OpenType SBIT format. Generally BitFonter cannot make changes to outline fonts, but when it saves a group of embedded bitmap fonts in Windows OpenType SBIT format it must make some necessary corrections. For example, if an opened project does not contain an outline component and is saved in OpenType SBIT (.ttf) format, BitFonter automatically generates **dummy outlines**. Further, you can choose what these generated glyphs will look like: spaces, dots, or squares. The **Always generate missing outline glyphs** option is always on.

The meaning of another option is illustrated in the following table:

Source font	Bitmaps	
	Embed bitmaps only in existing outline glyphs is ON	Embed bitmaps only in existing outline glyphs is OFF
with outline component	Extra bitmap glyphs are removed	All bitmap glyphs are saved*
w/o outline component	All bitmap glyphs are saved. Option is ignored	

* If the existing outline font has a glyph with more than one Unicode codepoint (and this is possible), only one corresponding bitmap glyph will be written and the others will be removed during the save operation.

The **Embed bitmaps only in existing outline glyphs** option is on by default.

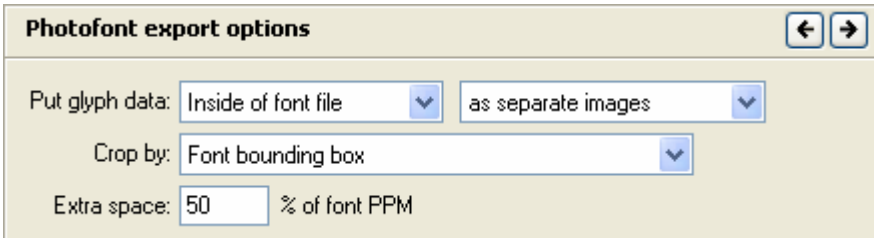
Other options are as follows:

... outline glyphs consisting of	When BitFonter generates missing outline glyphs, use this dropdown list to choose what these generated glyphs will look like: spaces, dots, or squares. Dots are the default selection
Generate bitmap scaling table	Use these two edit boxes to define the range of font sizes for which the embedded bitmap scaling table (EBSC) must be generated.

In order to better understand how bitmap scaling tables are generated let's look at the following example. Let's assume you have a Windows font set with two bitmap fonts: 10 and 14 pixels. In an OpenType SBIT font with embedded bitmaps you will get different scaled sizes depending on what values you enter in the **From** and **To** fields:

Values in the fields (From/to)	0/0	12/0	8/0	0/8	0/12	9/15	12/13
Sizes in EBSC table entries	11, 12, 13	12, 13	8, 9, 11, 12, 13	-	11, 12	9, 11, 12, 13, 15	12, 13
Sizes on the screen (real + scaled) in pixels	10, 11, 12, 13, 14	10, 12, 13, 14	8, 9, 10, 11, 12, 13, 14	10, 14	10, 11, 12, 14	9, 10, 11, 12, 13, 14, 15	10, 12, 13, 14

A zero value in the **From** field designates that the minimum scaling size will be taken from the smallest bitmap font in the family (10 pixels in our example). A zero value in the **To** field designates that the maximum scaling size will be taken from the largest bitmap font in the family (14 pixels in our example).



A Photofont (PhF font) may consist of one or several files: the main XML-based text file (which may also contain bitmap image data) and the bitmap image (or images) file containing actual glyph shapes referenced from the main file. I.e. the bitmap image data can be placed either in the main file or in separate files referenced by the main file.

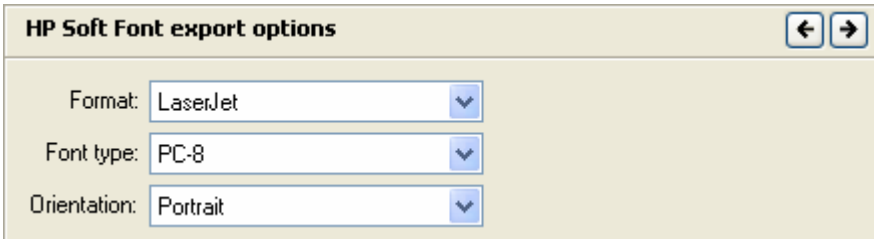
The first two dropdown lists control these glyph storage options. Choose **Inside of font file** if you want to keep glyph images in the main file or choose **Outside of font file** to keep them separately in .png files.

As separate images	Each glyph image will be saved separately (in the main .phf file or in several separate .png files). Separate files are placed in a new folder *fontname-Imgs*, where *fontname* is the name of the main phf file. Image file names have the following notation: glyphindex-glyphname.png
As single table image	Glyph images will be saved inside the main .phf file or externally in a .png file designed as a table of cells. One image – one cell
As single compact image	Glyph images will be saved inside the main .phf file or externally in a .png file designed to save space. Glyph images are placed very close to each other. The space between images can be changed with the **Extra space** parameter.

The **Crop by** dropdown list also has two options:

Font bounding box	Glyph images will be stored as a rectangles of one size equal to the size of the font bounding box (the white rectangle in the Glyph window)
Glyph's bounding box	Glyph images will be stored as rectangles of different sizes equal to each glyph's own bounding box. This option does not affect in situation when glyphs are stored as one image. The font bounding box is always used in such cases.

The **Extra space** parameter allows you to set the amount of space added at each side to the cropped glyph images.

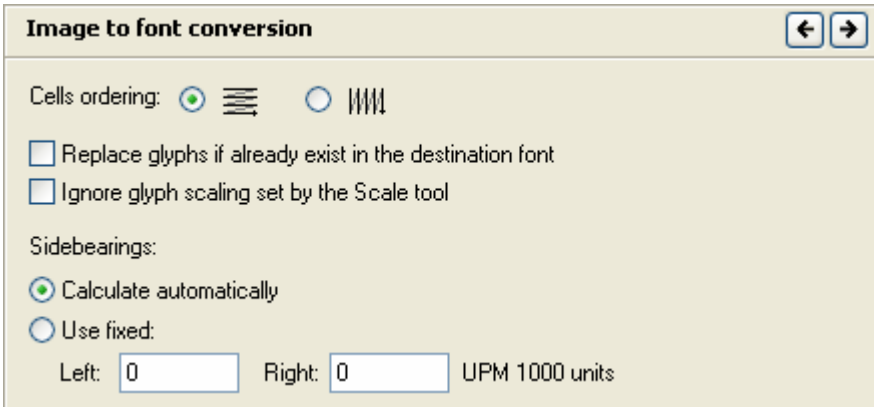


These options refer to saving HP Soft Fonts:

Format	Choose the type of printer for which your font is generated
Font type	Choose among 7-bit, 8-bit and PC-8 options
Orientation	Select Portrait or Landscape orientation (whichever the font is designed for)

Image To Font Conversion

The **Image To Font Conversion** page of the Tools > Options dialog allows you to customize the process of creating fonts from bitmap images in BitFonter:

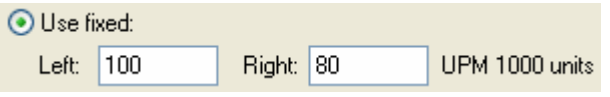


Define the order of placement of the image cells: horizontal or vertical. When **horizontal** order is selected, the glyph shapes will be taken from the image line by line, and when **vertical** order is checked, the shapes will be taken by columns.

When you place glyphs into the font you may overwrite existing glyphs. This situation can be controlled with the **Replace glyphs if already exist in the destination font** option. If it is switched on, the existing glyphs will disappear from the font. Otherwise, they will be moved to the end of the font.

The images are usually scaled when placed in the font. To not scale the glyph images, switch on the **Ignore glyph scaling...** option. The scaling factor set with the scale tool will be ignored.

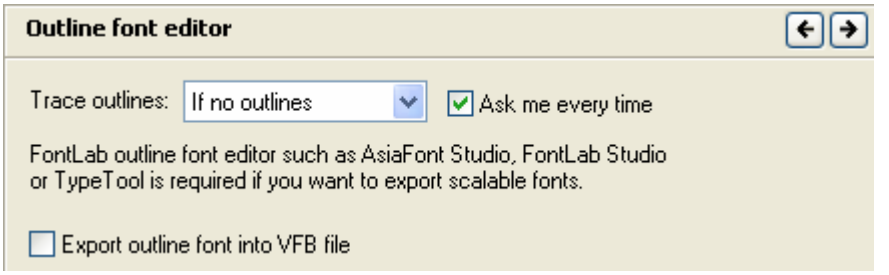
The **Sidebearings** section in the dialog allows you to define how BitFonter generates the advance width of glyphs. You can allow BitFonter to generate the left and right glyph sidebearings automatically or you may enter your custom fixed values:



Outline Font Editor

The **Outline Font Editor** page helps you customize the feature of BitFonter to communicate with outline font editors such as ***FontLab Studio*** (http://www.fontlab.com/studio/), ***AsiaFont Studio*** (http://www.fontlab.com/asiafontstudio/) or ***TypeTool*** (http://www.fontlab.com/typetool/).

This is useful for situations when you select the **Outline Font** or **Outline Pixelfont** command in the **File > Export** menu.


Outline font editor
Trace outlines:
If no outlines
Ask me every time
FontLab outline font editor such as AsiaFont Studio, FontLab Studio
or TypeTool is required if you want to export scalable fonts.
Export outline font into VFB file

The **Trace outlines** dropdown list allows you to control situations when your bitmap glyphs already contain an outline layer. If **Always** is selected BitFonter will ignore existing outlines. If **Never** is selected BitFonter will not trace but use only the existing outline layer. The default setting **If no outlines** means that only glyphs not having outlines will be autotraced but existing outlines will be placed into the output font without change.

If **Export outline font into VFB file** is not checked, the outline editor will be brought to the foreground after each **Export Outline Font** operation. Check this option if you want to save the output in the FontLab font format .vfb file.

Switch on the **Ask me every time** option to make BitFonter present the following dialog during each **Export Outline Font** operation (if any outlines present):

Export as Outline Font
Trace outlines if they're missing
Always trace outlines
Export existing outlines only
Trace options...
Always apply these settings for current document
If this option is on, hold down the Control key while choosing the menu command to open this dialog
Export
Cancel

Check the appropriate check box and proceed by pressing the **Export** button. Pressing **Cancel** will stop the **Export Outline Font** operation.

You can make these options permanent for the current font by setting the **Always apply these settings for current document** check box. In this case BitFonter will remember your choice and will not produce this dialog until you hold down the **Ctrl** key on the keyboard while choosing the **Export Outline Font** command.

If BitFonter cannot locate any outline font editor in your system or fails to communicate with a particular version of the server application, launch the application manually and try again.

Font Project

This chapter describes in detail how to manage fonts that you are editing or creating using Font Projects. Basically a font project is a file that contains references to real font files and families. It is important to understand that the project in native BitFonter format never contains a font itself; it keeps only aliases to real fonts and their properties on the disk. When you open e.g. a Microsoft Windows font set, it looks like the project in native BitFonter format too. But in this case it contains real fonts, not the aliases.

The BitFonter Standard Toolbar

Before we begin to work with projects we have to look at the BitFonter Standard toolbar:

The Standard toolbar contains command buttons for commonly used menu operations. All of them will be mentioned later in the manual, but we want to give their short description here:

	Creates new empty font in a new Font window
	Opens the font, image or the project from disk
	Saves the current font or the project to disk
	Cuts the selection
	Copies the selection
	Pastes the content of the Clipboard
	Undoes the previous operation
	Redoes the previously undone operation

If you want, you may choose to hide the Standard and other toolbars with the help of the **Toolbars** hierarchical menu in the **View** menu of the program.

Creating a New Project

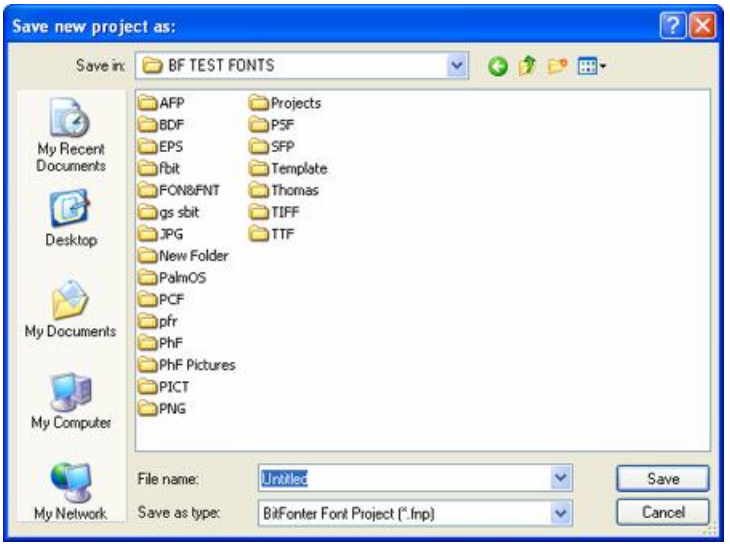
Font projects in BitFonter for Windows are of five types:

1. BitFonter native project format (.fnp extension)
2. Windows font set (.fon extension)
3. Windows OpenType SBIT font (.ttf extension)
4. Palm OS font (.pdb extension)
5. PhotoFont font (.phf extension)

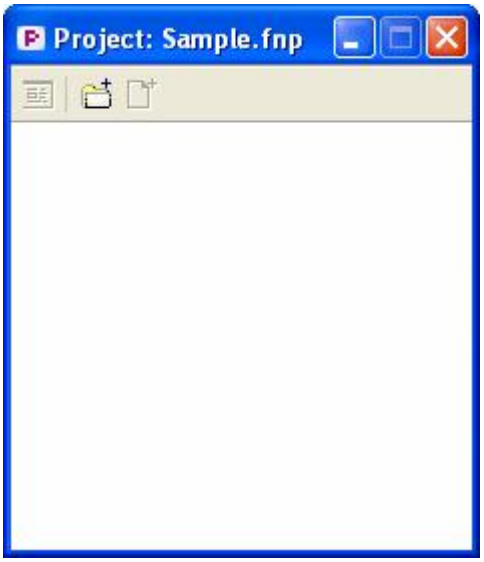
Before you create a new font project you must know what type of fonts you are going to manage. If you are not sure, create the project in the native BitFonter format.

To create a new project:

1. Select the **Project** command in the **New** submenu in the **File** menu. The standard Save File dialog will open:



2. Name your project, select the project format and destination and click on **Save**. We recommend you save the project in a new empty folder.
3. A new empty **Project window** will open:



Opening a Font Project

To open a font project for editing, select the **Open** command in the **File** menu or click on the button on the Standard toolbar if available. The standard Open File dialog appears. You will see all the files including image files that can be opened by BitFonter. If you want to list only files in a particular format, select the format in the **Files of type** dropdown list at the bottom of the Open File dialog. Select the project file you are going to edit and click on the **Open** button.

You can open the following types of file as a project:

1. BitFonter native project format (.fnp extension)
2. Windows font set (.fon extension)
3. Windows OpenType SBIT font (.ttf extension)
4. Palm OS font (.pdb extension)
5. PhotoFont font (.phf extension)

Or you also can open these files by dragging and dropping their icons onto the BitFonter's main window.

Working with Fonts in the Project Window

As was mentioned above, the project and its window contain a set of references to real fonts and their families stored separately or contain the fonts themselves. So working with the Project window you can manage the font sets easily but carefully.

The Project Window Toolbar

There is a Toolbar at the header of the Project window:

The Toolbar contains three buttons repeating main project related menu commands:

	Opens an Info dialog for the selected item
	Creates a new font family in the project
	Creates a new font and adds it to the project.

Font entries in the project and their icons

Every project is organized as a list of font families represented by a font family icon in the Project window. Usually a project contains one family but may contain as many as you want.

Every font family contains a set of font styles represented each by its own icon. These styles are Plain, Bold, Italic, and Bold Italic. Since every font has a style characteristic, every font of the project resides in its style folder.

Black&white bitmap fonts have a icon. Outline fonts are represented by a icon. Bitmap fonts checked as *Embedded* also have a icon. The embedded fonts are discussed in the ***Saving a Font Project*** (on page 81) section later in the chapter.

Color fonts naturally have color icons: fonts with 256 and fewer colors or levels of gray have a or icon, fonts with thousands or millions of colors have a icon.

➘ Note: In a Project window bitmap fonts are named by their *pixel* size not the point size.

Browsing the project font set

To view the contents of the font family, click on the triangle to the left of the family icon. You will see the list of font styles in the family. Click on the triangle to the left of the style icon to see the font of the particular style.

To select any font, click on its icon or name. To select multiple fonts click on their icons or names while holding down the **Shift** key. Selecting family or style selects all the enclosed fonts. You may also choose to select all the project fonts by the **Select All** command in the **Edit** menu.

You can move fonts from one style to another and even from one family to another by dragging their icons as you do with your files in Windows.

The Project Window context menu

If you right-click on the font in the Project window you will see the Project window context menu:

The menu contains several project related commands:

Create New Family	Creates a new font family in the project
Create New Font	Creates a new font and adds it to the selected style of the project
Font Info	Opens the Font Info dialog for the selected font
Verify	Verifies the project for errors
Delete	Deletes selected fonts or families from the project.

Creating new font families

To create a new font family, select the **Create New Family** command in the **Project** menu or click on the button located in the Project window toolbar. An empty font family named **Untitled** will be created. You can change the family name by editing its properties. The next new font family will get the name 'Untitled2' etc.

Creating new fonts and font references

To create a new font, select the **Create New Project Font** command in the **Project** menu or click on the button located in the Project window toolbar. You will see the New Font Dialog where you can define the new font parameters (see the ***Creating and Saving Fonts*** (on page 161) chapter for details on creating new fonts).

To create a new reference to a font (which already exists) in the project, drag the font to another family or style holding down the **Ctrl** and **Alt** keys. Be careful: this will copy the font's reference only but not the actual font. So, two lines in the project window will reference the same bitmap font. The reference font entry will get the icon and the referenced font entry name will be marked bold: **12 Pixels**.

This operation does not work with some types of projects like OpenType SBIT or PhotoFont format.

➘ Note: In fact when a font is "referenced" this means that only one real copy of the font is stored in the project. BitFonter highlights the first appearance of the font as "source" and others as "references". If you remove the "source" appearance from the project another appearance will take its place. To remove the real copy of the font from the project, you must remove all its references.

Duplicating, copying, and pasting fonts

You can duplicate, copy, and paste any font or even a font set. To duplicate fonts, select them by shift-clicking on their icons or names and then choose the **Duplicate** command in the **Edit** menu. Font duplicates will appear in the corresponding styles of the family. You may choose to copy fonts and paste copies to another style or another family. Use the standard **Copy** and **Paste** commands in the **Edit** menu or the corresponding buttons in the Standard toolbar. To paste a font to the particular style, select the style folder icon before pasting.

➲ Tip: It is possible to copy selected font(s) to another style or family by dragging while holding down the ALT key.

Deleting fonts and font families

There are two ways to delete selected items in the Project window. The first is to use the **Cut** or **Delete** command in the **Edit** menu or simply press the DEL key on the keyboard.

The second way is to select the **Remove from Project** command in the **Font** menu when the font is open in its Font window. See the ***Removing a Font from the Project*** (on page 147) section for details.

➘ Note: Be very careful when deleting fonts because this operation is not undoable. The only way to return the deleted font is to revert the whole project.

Viewing and editing family and font properties

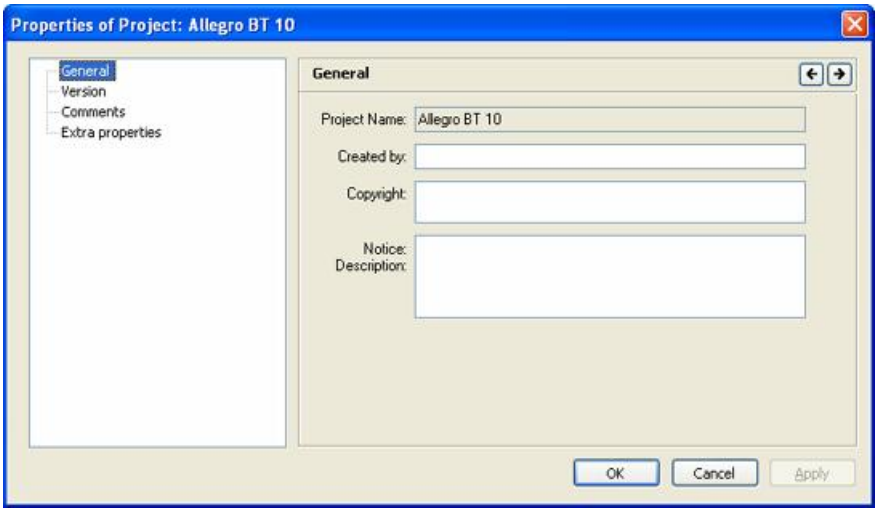
To view and edit the font family properties, select the family and choose the **Font Info** command in the **File** menu, or click on the button located in the Project window toolbar. The Family Info window will open (see the ***Editing Family Properties*** (on page 88) section in the ***Editing Fonts*** (on page 87) chapter for details).

To view and edit the font style properties, select the style slot and choose the **Font Info** command in the **File** menu, or click on the button located in the Project window toolbar. The Style Info window will open (see the ***Editing Style Properties*** (on page 89) section for details).

To view and edit the font properties, select the font and choose the **Font Info** command in the **File** menu, or click on the button located in the Project window toolbar. The Font Info window will open (see the ***Editing Font Properties*** (on page 90) section for details).

Font Project Info

While the Project window is open you may edit the project information. Choose the **Properties** command in the **Edit** menu. The Project Info dialog containing four pages will open:



Browse the pages using the list at the left or buttons.

To add new information to the project, click on the **Apply** button on any page. To apply information and close the Project Info dialog, click on **OK**. To close the dialog without saving changes, click on the **Cancel** button.

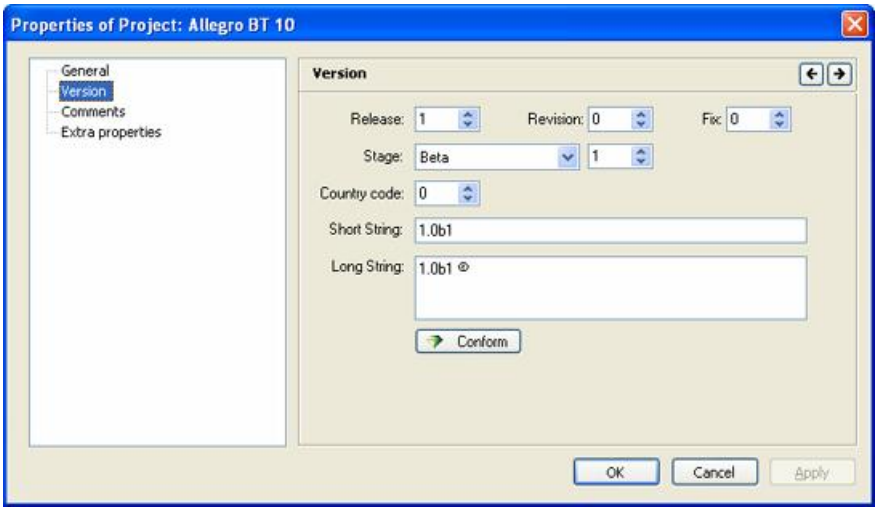
The Project Info fields are:

General page

Project name	Project title, which was filled in by BitFonter when the project was created. It must not be the same as the project file name
Created by	Name of the company or person that created the font project. If you are creating a new font project, enter your name or the name of your company here
Copyright	Copyright message. May include © sign or the word “Copyright”, the name of the company or person that owns the copyright and the copyright year
Notice	Additional information that you want to include in Project Info
Description	

Version page

Project version information will help you keep track of the work process:



Comments page

Any information you decide to provide with your project.

Extra properties page

The **Extra Properties** page is usually empty. Later it may contain some additional project information that cannot be interpreted by the current version of BitFonter.

All the fields in the Project Info dialog except the project name field are optional. You may leave them empty.

The information will be saved only in BitFonter's native project format file. This means that if you save the project, for example, as a Windows font, it will not contain any project info.

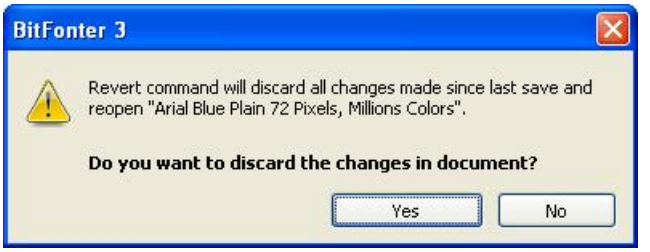
Verifying a Font Project

Sometimes while working with big and complicated projects you may need to verify them. Select the **Verify** command in the **File** menu while the Project window is active. The project will be checked for errors, and if there is one or more, you will see an Errors window with error descriptions. If there are no errors in the project, you will be informed with a 'The project's OK' message.

BitFonter verifies projects automatically when you select **Save**, **Save As** or **Save A Copy As** commands in the **File** menu. The project will not be saved if any error is found. You must correct any errors before saving again.

Reverting a Font Project

Sometimes you may need to revert the project to the last saved version. Reverting the project means reverting all its fonts even if they were saved before. To do this, select the **Revert** command in the **File** menu. You will see the revert confirmation dialog:


BitFonter 3
Revert command will discard all changes made since last save and reopen "Arial Blue Plain 72 Pixels, Millions Colors".
Do you want to discard the changes in document?
Yes
No

Click on the **Yes** button if you really want to discard all changes to the project. The last saved version of the project will be read from the disk in this case. Click on the **No** button if you have changed your mind.

Saving a Font Project

When you have finished working with your project and its fonts or want to save intermediate results, you should save the project.

To save the project to the same file (from where it was opened and in the same format), use the **Save** command in the **File** menu or simply click on the button on the Standard toolbar. For example, if you open a Windows font set (.fon) and make changes to it, the **Save** command will write changes to the same font set file.

➘ Note: When you save the project and there were unsaved open fonts all changes to the fonts will be saved automatically.

To save the project under a different name and/or in another format, you should use the **File > Save As** command. When you select this command, the standard Save As dialog appears. You can select the project format in the **Save as type** dropdown list located at the bottom of the dialog. Enter the project name and select the destination where you want to save your work. After the **Save As** command is successfully performed, the Project window will now contain this new font project with its new name/location.

To save a copy of the open project with a different name and/or in another format, use the **File > Save A Copy As** command. The difference between this command and the **Save As** command is that the original project remains open in its window.

BitFonter automatically verifies projects before saving them. The project will not be saved if any error is found. So you have to fix any errors before saving the project again.

You can save your project in the following formats:

1. BitFonter native project format (.fnp extension)
2. Windows font set (.fon extension)
3. Windows OpenType SBIT font (.ttf extension)
4. Palm OS font (.pdb extension)
5. PhotoFont font (.phf extension)

Saving BitFonter's Project File

A project file in native BitFonter format (.fnp extension) never contains a font itself. It keeps only aliases of the real font files and their properties. Fonts themselves are stored in separate files on the disk. So the result of saving in this special format may vary depending on the command you use for saving. The native format project may contain one or several families of fonts. These families may contain aliases both to the separate fonts on the disk and to whole families or font sets.

If you choose the **File > Save As** command to save in native project format, only the project file with aliases inside will be written to the disk. The real fonts referred to by the aliases will remain at their old locations, wherever they were. They are "linked" to the project. So, be attentive when saving with the **Save As** command.

If you want to transfer your work to another computer, it is better to use the **File > Save A Copy As** command. When you choose the **Save A Copy As** command all the fonts that reside in e.g. TrueType/OpenType TT fonts or Windows font sets and that are referred to by the project will be saved separately in the default font storage format along with their project file. Initially the default font storage format is BFB (BitFonter Document) format, but you can change it in the Tools > Options dialog described in the ***BitFonter Options*** (on page 44) section. The fonts that have not been combined into sets and reside separately somewhere on the disk will be saved along with their project file too. As a result you will get a project file with all its fonts gathered in one place. The original project remains open in its window. This command is similar, for example, to the **Collect for Output** command in QuarkXPress.

Saving a Windows Font Set

A Windows font set may contain one or several families of fonts. Usually one font set file contains one family, but this is not necessary. The fonts are classified in the family by style and size. There cannot be two fonts of the same size and style in one family . These fonts will be treated as duplicates and the project will not be saved. Every font in a font set must meet the general requirements of a Microsoft Windows font. See the ***Saving in Windows Font Format*** (on page 207) section in the ***Creating and Saving Fonts*** (on page 161) chapter for details. You will not be able to save the whole font set if you have not saved the enclosed fonts properly first.

The Windows font set cannot contain outline fonts.

To save a font project in Windows Font Set format:

1. Save any of the project's fonts that have not already been saved.
2. Make sure there are no outline fonts in the project and that there are no duplicate bitmap fonts that are of the same size and style.
3. Select the **Save As** command in the **File** menu; select Windows Font Set in the **Save as type** dropdown list of the Save As dialog; name the file; select the destination and click on the **Save** button.

Saving a Windows OpenType SBIT Font

A project containing only one family can be saved as a Windows OpenType SBIT font. All bitmap fonts in the project must refer to the one style. If the project contains non-embedded bitmap fonts they will be checked as embedded automatically during the save operation. A project containing duplicate bitmap fonts (two fonts of the same size and color mode) cannot be saved as a Windows OpenType SBIT font.

Windows OpenType SBIT font may contain only embedded bitmap fonts and BitFonter will create an outline component automatically. Every embedded font must meet the general requirements of a Microsoft Windows font. See the ***Saving in Windows Font Format*** (on page 207) section in the ***Creating and Saving Fonts*** (on page 161) chapter for details. Besides meeting the general requirements such a font must also have Unicode codepoints and may be of 4 or 16 levels of gray with the corresponding Windows color tables (see the ***Saving Color Fonts*** (on page 210) section). Also refer to the **Export > OpenType SBIT** page of the Tools > Options dialog to customize the OpenType SBIT font export options described in the ***BitFonter Options*** (on page 44) section.

➘ Note: When OpenType SBIT font with embedded bitmaps only is generated all the font metrics are taken from the font of the largest pixel size.

To save the font project in Windows OpenType SBIT format:

1. Save any of the project's fonts that have not already been saved. Customize the OpenType SBIT export options in the Tools > Options dialog.
2. Make sure there is only one family in the project, that the bitmap fonts belong to one style, and they are of the right color type. Note that you do not have to check bitmap fonts as embedded when saving into OpenType SBIT format - all bitmap fonts are embedded into an OpenType SBIT font by default.
3. Select the **Save As** command in the **File** menu; select OpenType SBIT Font in the **Save as type** dropdown list of the Save As dialog; name the file; select the destination and click on the **Save** button.

Saving a Palm OS Font

A project containing only black & white bitmap fonts can be saved as a Palm OS font. If the project contains at least one outline font, it cannot be saved as a Palm OS font file.

Every bitmap font must meet the following requirements of a Palm OS font: it cannot have more than 256 glyphs, it cannot have glyphs with duplicate codes, and it cannot have glyphs larger than 256x256 pixels. The glyphs' names and Unicode codepoints are not written in this format. Nor is any general font information saved. Saving in this format entails a great loss of information. (So you might want to save in another format first.)

To save the font project in Palm OS font format:

1. Save any of the project's fonts that have not already been saved with the **File > Save** command.
2. Make sure there are no outline fonts in the families, and that the bitmap fonts are of the right color type (black & white).
3. Select the **Save As** command in the **File** menu; select Palm OS Font in the **Save as type** dropdown list of the Save As dialog; name the file; select the destination and click on the **Save** button.

➘ Note: Although you do not check bitmap fonts as embedded when saving for Palm OS, all bitmap fonts will be checked as embedded automatically.

➘ Note: The pixel size of a Palm OS font is always defined by font ascent.

Saving in PhotoFont Format

A font in PhotoFont (PhF) format contains glyphs as a set of true color bitmap images. You can save any existing font of any bitmap format in this format. A font containing only TrueType/OpenType TT outlines or fonts of more than one style cannot be saved as a PhotoFont font.

There are no restrictions on the glyphs in this format but one: a font of any color mode will be saved as a 32-bit true color font. This is the reason the .phf font file is 10 times bigger than its black & white twin. The PhotoFont format accepts glyphs' names, codes, and Unicode codepoints but they are not necessary. All these features make this format very useful in graphic design, desktop publishing, creating presentations and web authoring.

The PhF font may consist of one or several files: the main XML-based text file (which may also contain bitmap image data) and the bitmap image (or images) files containing actual glyph shapes referenced from the main file. I.e. the bitmap image data can be placed either in the main file or in separate files referenced by the main file. Refer to the **Export > Photofont** page of the Tools > Options dialog to customize the PhotoFont export options described in the ***BitFonter Options*** (on page 44) section.

To save the font project in PhotoFont format:

1. Save any of the project's fonts that have not already been saved. Customize the PhotoFont export options in the Tools > Options dialog if necessary.
2. Make sure there are no outline fonts in the families.
3. Select the **Save As** command in the **File** menu; select PhotoFont in the **Save as type** dropdown list of the Save As dialog; name the file; select the destination and click on the **Save** button.

Editing Fonts

Any font can be modified at four different levels. They are:

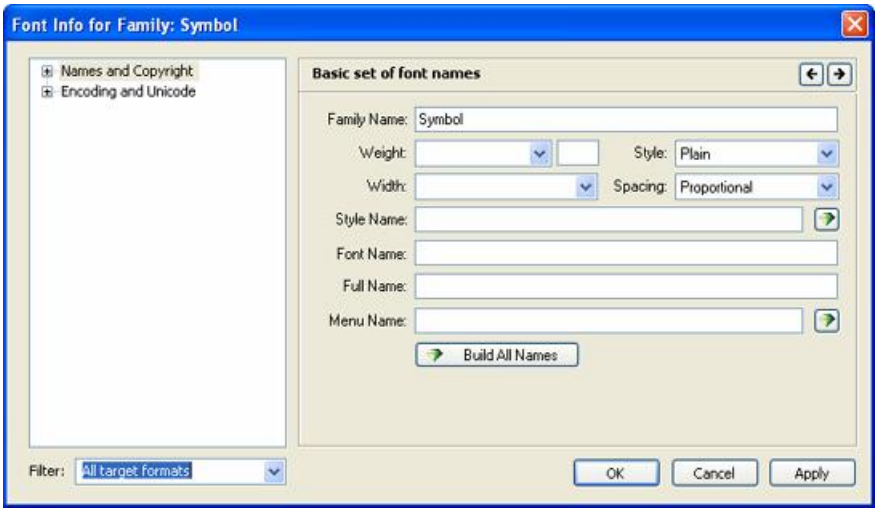
1. Font family and style properties level
2. Font properties level
3. Font character set level
4. Glyph level

The first three levels are the subject of this chapter. The last and the most complicated level will be described in the ***Editing Glyphs*** (on page 225) chapter.

First we are going to review font family and style properties editing, then we will refer to individual font modifications.

Editing Family Properties

To view or edit font family properties, select the family icon in the Project window and choose the **Font Info** command in the **File** menu, or click on the button located in the Project window toolbar. You will see the Family Info dialog:



The Family Info dialog contains five pages. Browse the pages using the list at the left or the buttons.

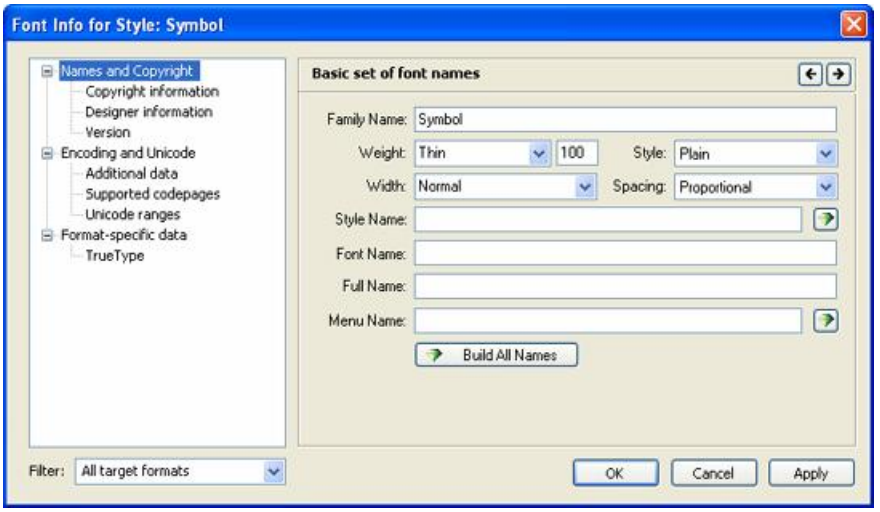
To apply new information to the font family, click on the **Apply** button on any page. To apply information and close the Family Info dialog, click on **OK**. To close the dialog without saving changes, click on the **Cancel** button.

In the ***Editing Font Properties*** (on page 90) section we will describe all the Info pages and give some recommendations for entering data properly.

Editing Style Properties

The style properties are the font style information that can be applied to fonts in a very few cases. Although BitFonter cannot edit outline fonts (neither Type 1 fonts, nor TrueType/OpenType TT), it can write an OpenType SBIT font in Windows format with embedded bitmap fonts in it. In this case the style properties may be useful.

To view or edit the style properties, select the style slot in the Project window and choose the **Font Info** command in the **File** menu or click on the button located in the Project window toolbar. The Style Info dialog will appear:



The Style Info dialog is a subset of the Font Info dialog and contains only pages with fields relevant for styles and families. Browse the pages using the list at the left or buttons.

To apply new information to the style, click on the **Apply** button on any page. To apply information and close the Style Info dialog, click on **OK**. To close the dialog without saving changes, click on the **Cancel** button.

In the ***Editing Font Properties*** (on page 90) section we will describe all the Info pages and give some recommendations for entering data properly.

➲ Note: The Style Info dialog always includes the Family Info pages.

Editing Font Properties

Font properties is a collection of information that defines a font. The best example of this information is the Font Name or Font Style. Font Info data also includes copyright information, identification information, font dimensions and other data. Much of this information can be applied to families and styles.

When you open an existing font, the Font Info fields are filled by the information from the font file. Missing information is automatically calculated where possible.

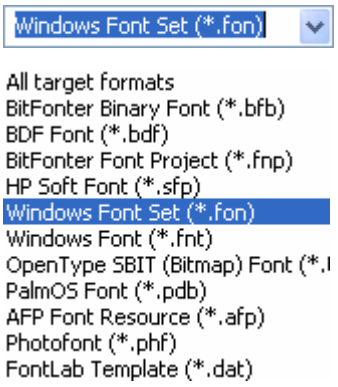
When you create a new font, you must provide some information in the New Font dialog. BitFonter fills in other font information automatically. To get an accurate font that will work everywhere, you have to check all these values and enter corrected data if necessary.

To open the Font Info dialog, select the font in the Project window and choose the **Font Info** command in the **File** menu or click on the button located in the Project window toolbar.

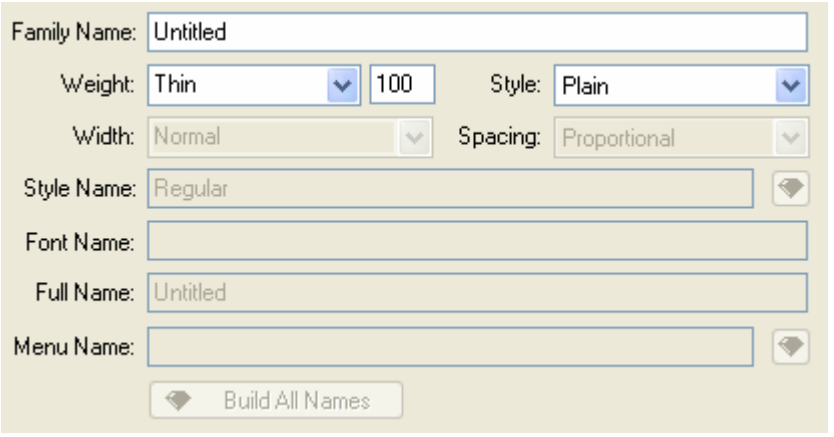
The Font Info dialog looks like Family Info and Style Info dialogs and includes the complete set of pages. Indeed the Family Info and Style Info pages are the subset of the Font Info pages.

Browse the pages using the list at the left or buttons.

Use the **Filter** dropdown list at the bottom of the Font Info dialog to disable those fields and options that are not relevant for this or that font format:



For example, if you select *Windows Font Set (*.fon)* as a filter you can see the **Basic Set of Font Names** page changes its view:

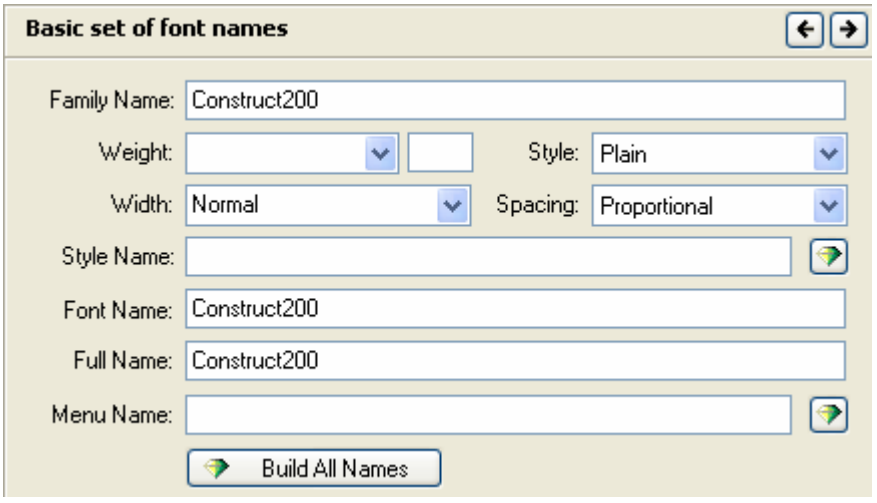


Only the **Family Name**, **Weight** and **Style** are relevant for a font in the selected format.

To apply new information to the font, click on the **Apply** button on any page. To apply information and close the Font Info dialog, click on **OK**. To close the dialog without saving changes, click on the **Cancel** button.

Basic Set of Font Names

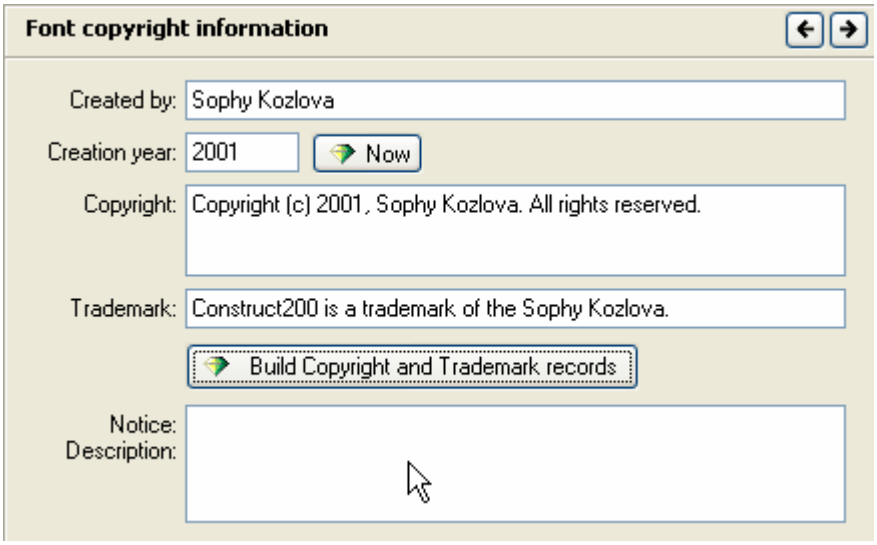
The **Names and Copyright** page includes the most general information applicable to families, styles and fonts:



Family Name	The name of the typeface to which all the enclosed fonts belong. The family name is the name that appears in an application's font menu. It may differ from the name of the file containing the family. So be careful when you rename font families. Do not mix up the Family Name and the Project Name. Although these names may be the same these are quite different things, since one project may contain several families
Weight	Weight of the font. You may enter a custom value in this field or select one of the predefined weight names in the dropdown list. Values in this list are sorted by increased weight value. Choose Normal or leave this field empty if you do not care about the font's weight
Width	The average advance width of the font's glyphs. Enter a custom value or select one of the predefined width values from the dropdown list. Leave this field empty or select Normal width if you do not care about the font's width
Style	The overall posture of the typeface design used in the font. Select Italic if you are creating italic font or Roman (upright design) in other cases
Spacing	Font attribute that defines advance widths of glyphs. In Proportional fonts, advance widths vary for each glyph and are defined by the left and the right margins of glyphs. A Monospaced font is a font whose advance widths are the same for all glyphs (defined as Average Width). The Char Cell attribute defines a monospaced font whose glyphs have the same glyph width and height — usually defined by the font bounding box
Style Name	Contains complete style information about the font. We recommend that you fill in the **Weight**, **Width** and **Style** data, to automatically generate this field using the **Build** button located to the right and edit this field if necessary
Full Name	More detailed font name. May include spaces as well as any other characters
Font Name	Complete font name. This name will appear in the font selection menu and dialog boxes of applications that use fonts. Do not include spaces
Menu Name	The name used to access a font in applications. This name must not include style information (bold, italic or similar). The length of this field is limited to 27 characters. To control that the current Menu name is made properly, click on the **Check&Fix** button
Build All Names	Click on this button to automatically generate all Name fields. If you are creating a new font we recommend that you fill in the **Family Name** field in the Family Info dialog, generate or manually fill in the **Style Name** field and press this button to create the Font and Full names. If necessary you can edit the names later.

Copyright Information

The **Copyright Information** page of the Font Info dialog contains data about the font creators, the font trademark name, copyright owners etc. and looks like this:



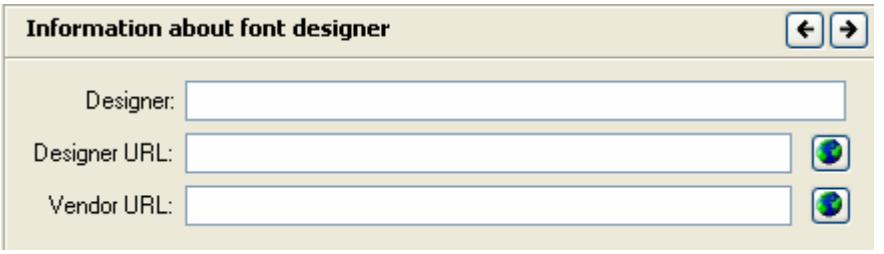
On this page you can enter information about the creators of the font. If you have created a new font, you should enter your copyright notice here. If you have edited an existing font that was not your creation you must not change the information contained on this page or you may violate copyright laws:

Created by	Name of the company or person that created the font. If you are creating a new font, enter your name or the name of your company here
Creation year	Year when the font was created. This is used by BitFonter to automatically fill in the Copyright field
Copyright	Copyright message. Must include © sign or the word "Copyright", the name of the company or person that owns the copyright and the copyright year
Trademark	Font trademark — used to save font's trademark notice
Notice	Additional information that you want to include in Font Info
Description	
Build Copyright and Trademark records	Press this button to create the standard Copyright record based on the Created by and Creation Year fields

To automatically generate the **Copyright** and **Trademark** fields using information from the **Created by** and **Creation Year** fields, click on the **Build Copyright and Trademark records** button.

Designer Information

The **Designer Information** page of the Font Info dialog contains data about the font designer and looks like this:



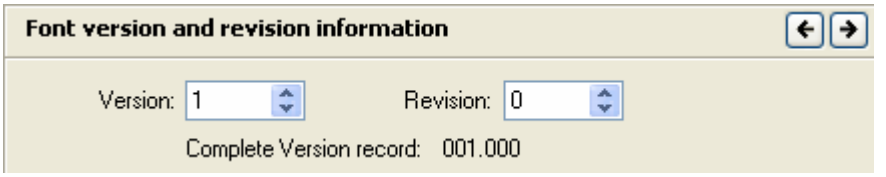
This page stores information about the font's designer. Do not modify this data if you open an existing font to modify for personal use.

Designer	Name of font designer
Designer URL	A new entry implemented only in TrueType/OpenType TT format. It is the WWW link to the designer of the font
Vendor URL	This TrueType-only entry shows the WWW link to the site of the font vendor.

Use the buttons to the right of the **Designer URL** and **Vendor URL** controls to open pages in a Web browser window. This requires an Internet connection.

Version Information

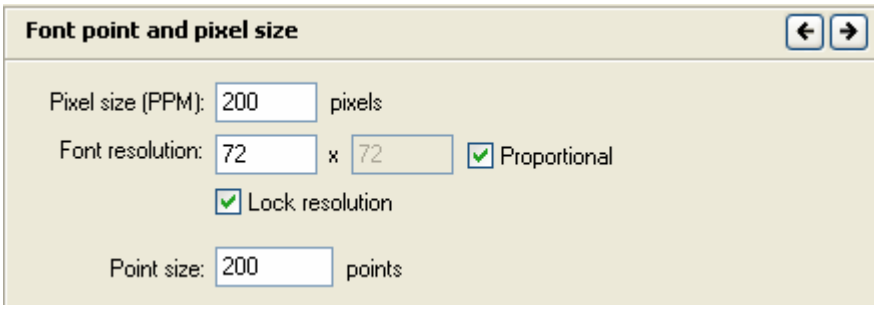
The **Version** page of the Font Info dialog contains data about the font version:



Version	Version of the font.
Revision	Revision of the font. Version and revision numbers are combined and build a complete version record

Metrics and Dimensions

The **Metrics and Dimensions** page contains the most general font metrics information and looks like this:


Font point and pixel size
Pixel size (PPM): 200 pixels
Font resolution: 72 x 72 Proportional
Lock resolution
Point size: 200 points

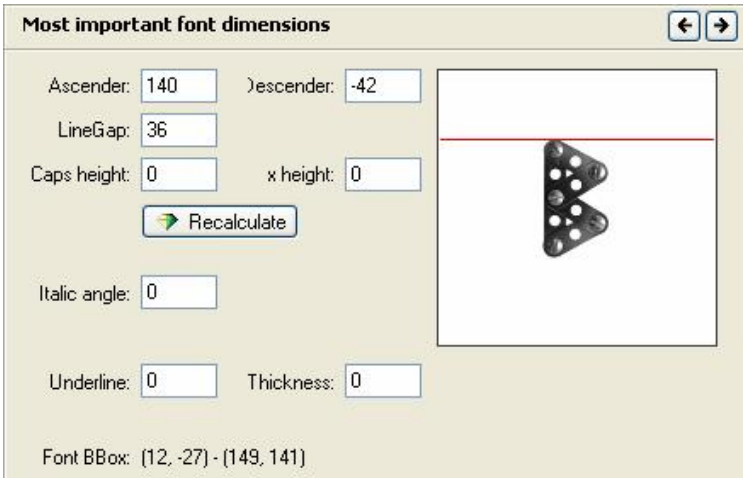
There are a several metric fields on this page:

Pixel size (pixels)	This is the point size in pixels calculated from the font resolution
Resolution	The horizontal and vertical font resolution in pixels per inch (72 ppi is usual for Macintosh screen fonts)
Point size (points)	This is the designed font size in points. It is calculated from the Pixel size and Resolution.

When checked, the **Lock resolution** option locks the font resolution making the **Point size** field recalculated while you change the **Pixel size** field and vice versa.

Key Dimensions

The **Key Dimensions** page contains the most important font metrics information including its vertical metrics:



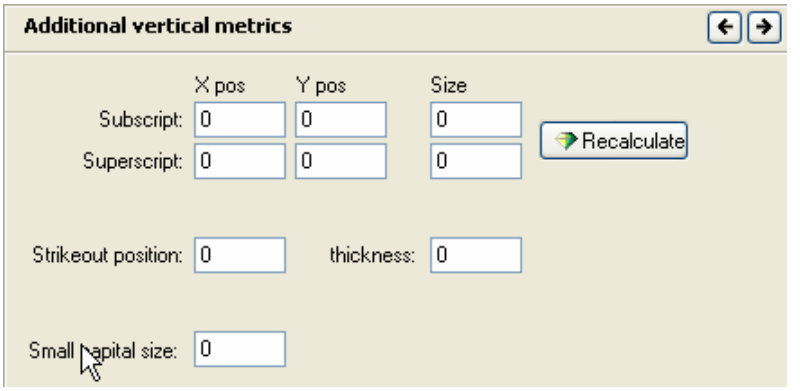
There are a several metric fields and a preview panel on this page:

Ascender	Position of the font's ascender line in pixels. Usually this is the position of the top line of the lowercase 'b' glyph
Descender	Position of the font's descender line in pixels. Usually this is the position of the bottom line of the lowercase 'g' glyph
LineGap	This is the distance between two lines of text - between the descender of the first line and the ascender of the next line.
Caps height	Height of the font's uppercase glyphs. Usually the height of the 'H' glyph
x height	Height of the lowercase glyphs. Usually the height of the 'x' glyph
Italic angle	Angle for slanting the glyphs when automatic italicization is made. Usually not used
Underline	The recommended vertical offset in pixels from the baseline to the top of the underline. It is used for text underlining
Thickness	This is the thickness of the underline line in pixels.

If you click on the **Recalculate** button, BitFonter will automatically recalculate the values above the button.

Additional Vertical Metrics

The **Additional Vertical Metrics** page contains additional font metrics information and looks like this:



Most programs do not use this information, so it may be undefined in your font. Click on the **Recalculate** button to get the default automatically calculated values.

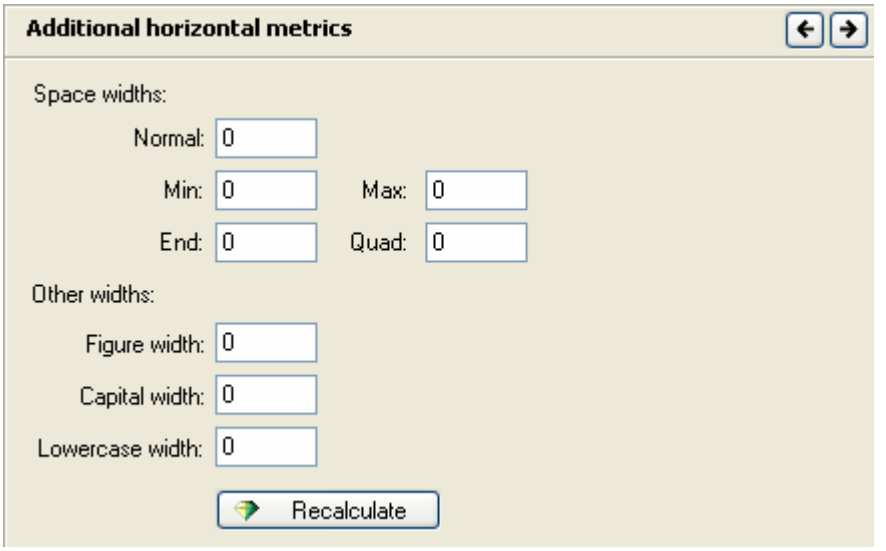
Here is the short description of the fields:

Subscript X offset	The recommended horizontal offset in pixels to the X origin of synthetic subscript text
Subscript Y offset	The recommended vertical offset in pixels to the Y origin of synthetic subscript text
Subscript size	The recommended body size of synthetic subscripts to be used with this font, in pixels.
Superscript X offset	The recommended horizontal offset in pixels to the X origin of synthetic superscript text
Superscript Y offset	The recommended vertical offset in pixels to the Y origin of synthetic superscript text
Superscript size	The recommended body size of synthetic superscripts to be used with this font, in pixels.
Strikeout position	The recommended vertical offset in pixels from the baseline to the top of the strikeout line

Thickness	This is the thickness of the strikeout line in pixels
Small capital size	The recommended body size of synthetic small capitals to be used with this font, in pixels. Small capitals are generally imaged from a smaller font.

Additional Horizontal Metrics

The **Additional Horizontal Metrics** page contains horizontal font metrics information:



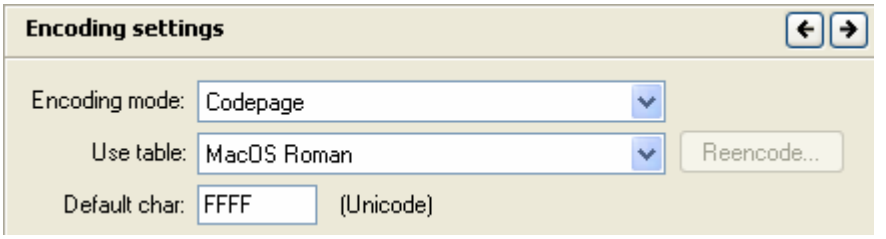
Most programs do not use this information, so it may be undefined in your font. Click on the **Recalculate** button to get the default automatically calculated values.

Here is the short description of the fields:

Normal space	The recommended normal word-space value to be used with this font. If the 'space' glyph exists in the font its advance width will be the normal space
Minimum space	The recommended minimum word-space value to be used with this font. The minimum space is usually 0.75 of the normal space
Maximum space	The recommended maximum word-space value to be used with this font. The maximum space is usually 1.5 of the normal space
End space	The recommended spacing value at the end of sentences. The end space is usually equal to a normal space
Quad space	This metric gives the advance width of a quad (em) space. The use of this font property has been deprecated
Figure width	The advance width of the tabular figures and the dollar sign if all advance widths are equal. For Latin fonts, these tabular figures are the Arabic numerals '0' through '9'
Capital width	The arithmetic mean of the absolute value of the advance width of each capital glyph in the font in pixels. For Latin fonts, capitals are the glyphs 'A' through 'Z'. This property is usually used for font matching or substitution
Lowercase width	The arithmetic mean of the absolute value of the advance width of each lowercase glyph in the font in pixels. For Latin fonts, lowercase are the glyphs 'a' through 'z'. This property is usually used for font matching or substitution.

Encoding and Unicode

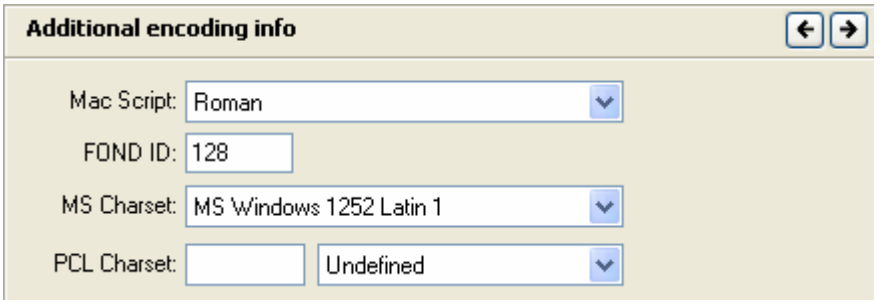
The **Encoding and Unicode** page includes the most general information that is applied to all the fonts in the family:



Encoding mode	The Font window editing mode which is used here as a font export setting. Use this and the next field to define the font export encoding
Use table	The encoding table of the font based on the Font window editing mode. The contents of this dropdown list depend on the selection in the **Encoding mode** dropdown list above
Reencode	This button runs the automatic font reencoding procedure. See the ***Reencoding Glyphs Automatically*** (on page 140) section for details
Default Char	The Unicode codepoint of the font default character. The default character is used in text for showing codes not represented in the font character table by any glyph.

Additional Encoding Info

The **Additional Encoding Info** page includes some more rarely used encoding information:

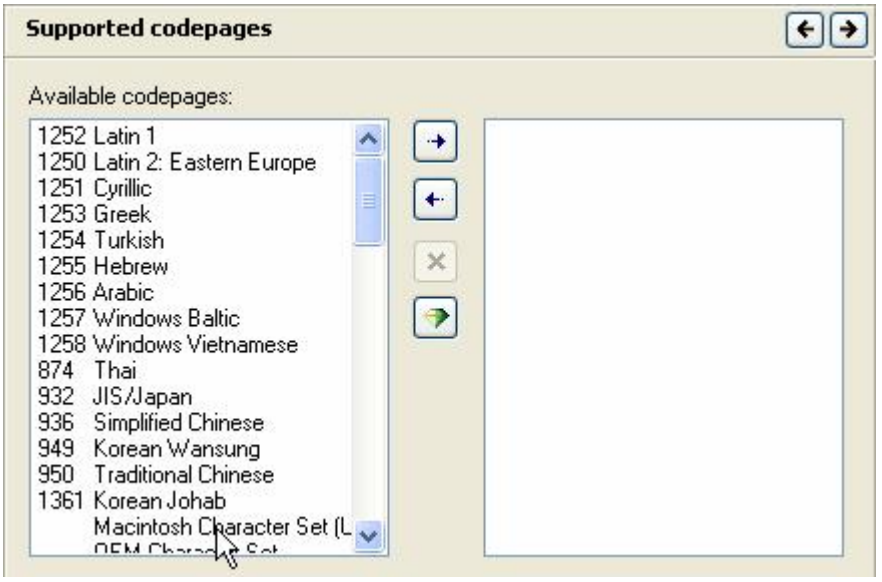


Mac Script	The scripting system for Macintosh fonts. The selection in the **Script** dropdown list automatically sets the **FOND ID**. It is of no use if you are working with a non-Macintosh font
FOND ID	The identifier of the 'FOND' resource in a Macintosh font. It is set manually or automatically when you select the script. It is of no use if you are working with a non-Macintosh font
MS Charset	Microsoft Windows Character Set must be defined if you are working with a Windows font
PCL Charset	If you are going to generate a HP Soft font, type in or select in the dropdown list the appropriate PCL Character Set.

When saving the font to a specific format, only the corresponding information will be used and saved. In other words, you need not, for example, specify PCL Character Set, if you are not going to save the font as a HP Soft font.

Supported Codepages

As you know, fonts may have very many characters and support a lot of different languages. To give the operating system information about what languages the current font can support, you have to set the codepages information. The part of this information is set on the **Additional Encoding Info** page . Also, for OpenType SBIT fonts you can identify all the supported codepages:



There are two lists of codepages in the dialog: the **Available** codepages to the left and the **Supported** codepages to the right. To find out what a codepage is, see the ***Codepages*** (on page 123) section.

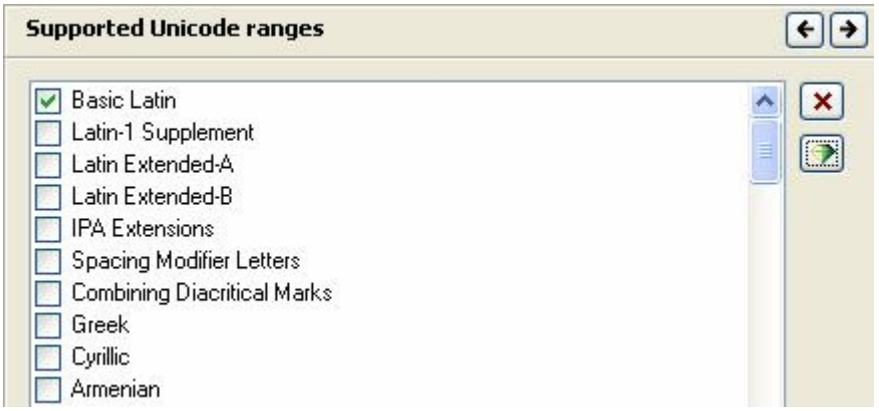
To add a codepage to the list of supported codepages, select a codepage in the left list and click on the **Add** button.

To remove a codepage from the list of supported codepages, select a codepage in the right list and press the **Del** button.

To reset the list of supported codepages, click on the **Reset** button.

To select the supported codepages automatically, click on the **Auto** button. BitFonter will analyze the Unicode information available in the font and will automatically detect which codepages this font can support.

Supported Unicode Ranges



OpenType fonts must declare Unicode ranges that the font can support so that the operating system can decide which characters the font can be used to represent.

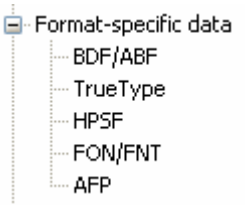
The **Supported Unicode ranges** page is relatively simple: you can see a list of all Unicode ranges with a check box to the left of each name. If the check box is checked it means that range is supported by the font.

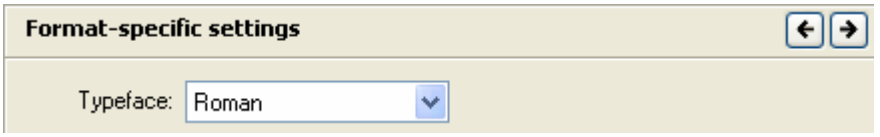
Buttons to the right of the list mean:

✕	Uncheck all ranges
	Automatically check ranges using information about Unicode codepoints assigned to fonts glyphs.

Format Specific Data

The **Format-specific data** section contains more sometimes useful font information specific for different font formats:

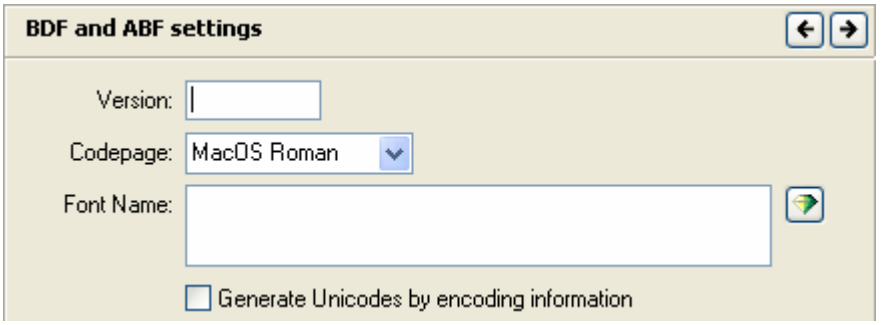




The **Typeface** dropdown list allows you to set the overall posture of the typeface design used in the font. It is usually used in BDF fonts. Select *Italic* if you are creating an italic font or *Roman* (upright design) in other cases.

You do not need to fill in all the fields on the pages that follow. For example, fill in the fields on the **FON/FNT** page only if you are going to save in the Windows Font Set format etc.

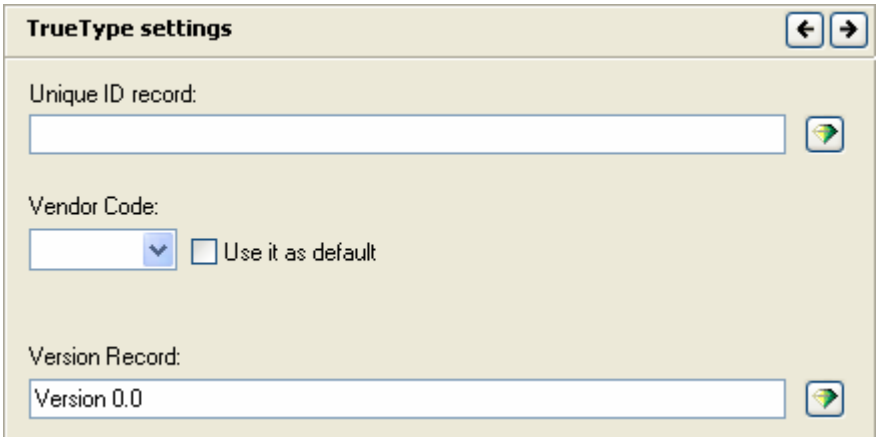
BDF and ABF Data



Here is the short description of the fields:

Version	Any string representing the formal or informal version of the font. It is valid only for BDF format
Codepage	Codepage information used only in the BDF **Font Name** field
Font Name	Full font name for use in BDF format fonts only. Click on the corresponding **Build** button to generate a correct name.

TrueType Data

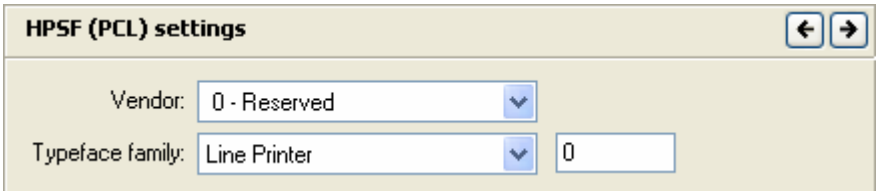


To make a proper font you may simply click on the (**Build**) button where applicable.

Here is the short description of the TrueType/OpenType TT specific fields:

Unique ID record	This field is necessary to identify TrueType/OpenType TT fonts. Usually it includes the creator's name, family name and creation year. The format of this field is freeform, but we recommend that you use the Build button to fill this field automatically
Vendor code	An up-to-four letter code that is assigned to most TrueType producers to identify their fonts. An *uppercase* vendor code must be registered with Microsoft or Apple. All registered Vendor codes known at the time of BitFonter's release are placed in the dropdown list. If you want to identify yourself without registering you may enter a *lowercase* four-letter vendor code
Version record	TrueType/OpenType TT font version records have a different format. You may enter the TrueType/OpenType TT version record here or click on the **Build** button at the right of the field to fill this record automatically.

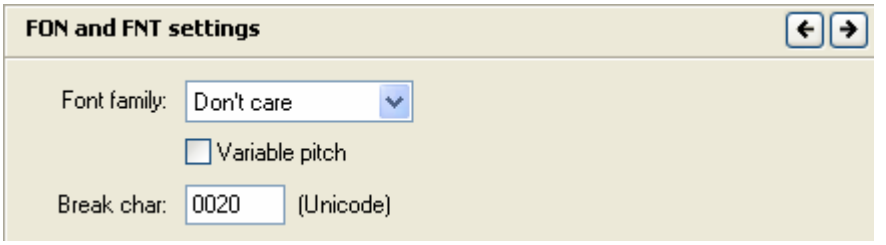
HPSF Data



Here is the short description of the HPSF (PCL) fields:

Vendor	The vendor code for the HP Soft font
Typeface family	If you are going to generate a HP Soft font, type in or select in the dropdown list the appropriate **PCL Typeface Family.**

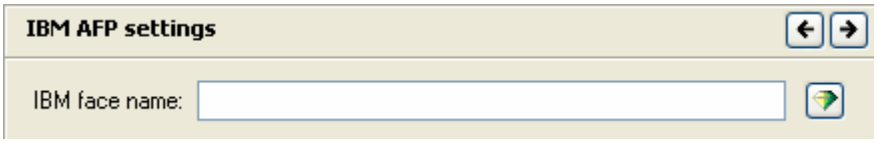
FON and FNT Data



Here is the short description of the fields used in FONs and FNTs:

Font Family	Font families describe in a general way the look of a font. They are intended for fonts when the exact face name desired is not available
Variable Pitch	Variable pitch means proportional. This option should be off for a monospaced font
Break Char	This Unicode value specifies the character that will define word breaks. This character defines word breaks for word wrapping and word spacing justification in FON and FNT fonts. It is the 'space' character (0x0020) usually.

AFP Data



The **IBM Face Name** is used in AFP files only. You cannot automatically generate this name in BitFonter but you may enter it manually if needed.

Extra Properties and Font Note Pages

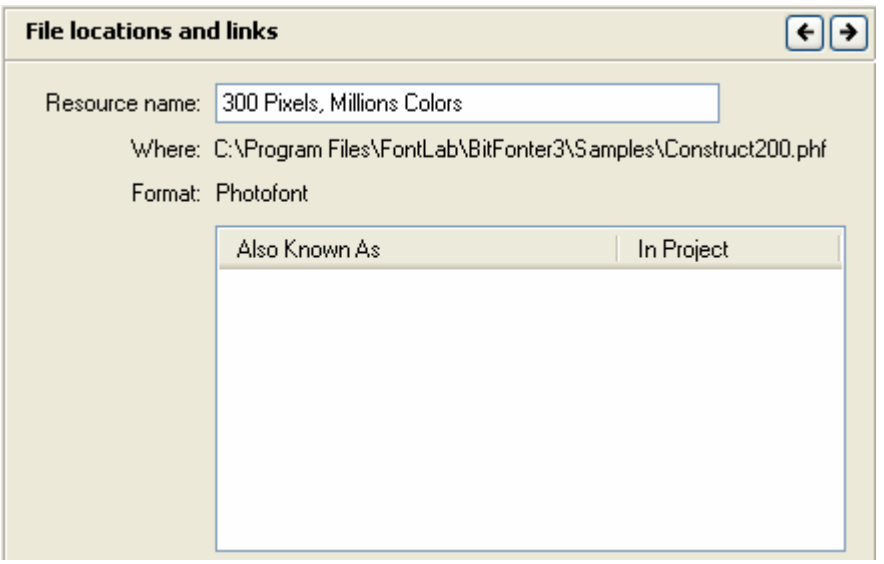
The **Extra Properties** page is usually empty. In future versions it may contain some additional font information.

The **Font Note** page can be used for adding any text information you want while working with your font. But note that this information will be written to BDF format file only.

File Location and Links

The **File Location and Links** page includes the following font information:

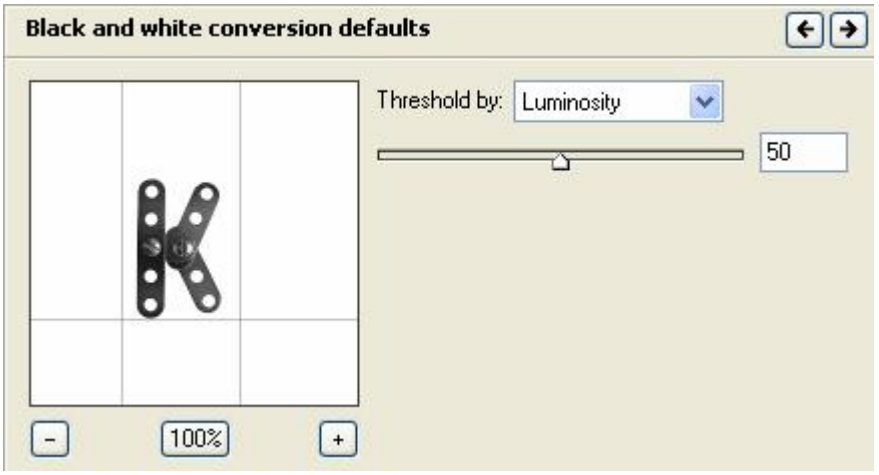
Resource Name	BitFonter names bitmap fonts on the basis of their pixel size. You can change the font name in the name field. Type any name you want but note that this name has nothing in common with the family name and will be written in the native BitFonter project format file only.
Where	This field shows you the font or its container location on the disk
Format	The format of the font or its container.



If the font belongs to a project or projects, you can see this in the project list.

Black/White Conversion

If you are going to make a black & white font from a color or grayscale font, you usually have to define the threshold parameter for conversion. The threshold parameter tells BitFonter what pixels are light and what pixels are dark. Light pixels of the glyph image are replaced with white color and dark pixels with black. The boundary between "light" and "dark" can be defined here on the **Black/White Conversion** page:



The conversion can be performed by one of the following parameters:

Luminosity	The luminance countered from pixel's red, green and blue channels
Opacity	Pixel's opacity/transparency component (for fonts with millions of colors)
Color	Pixel's hue angle.

To see the original glyph in the preview, press and hold down the **Ctrl** key.

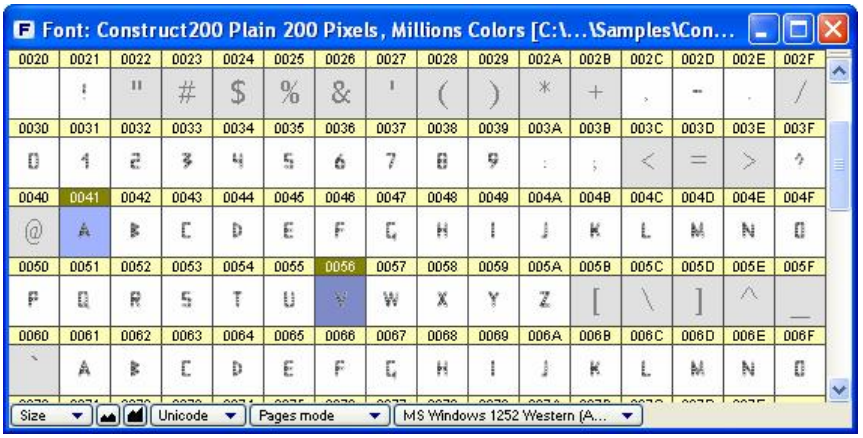
Editing a Font Character Set

When speaking about editing a character set we mean the process of editing font's glyphs' attributes, not editing glyph shapes. Glyph editing techniques will be described later in the ***Editing Glyphs*** (on page 225) chapter.

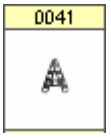
Font Window

The Font window is used to display an entire font. It opens automatically when you open an existing font by double-clicking on its icon in the Project window or with the **File > Open** command. You can also create a new font with the **New Font** command in the **File** menu or with the **Create New Project Font** in the **Project** menu or by importing a font.

A Font window containing all the font glyphs will open:



The Font window consists of the *footer* at the bottom, and a glyph *table,* where a single cell represents each glyph:

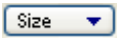
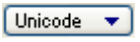


Each cell has a *caption* at the top that shows some identification information — it may be the name of the glyph, its code in various forms or some other glyph information.

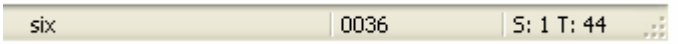
➲ Tip: You can change the font size used to display the caption in Tools > Options **> Font Window > View**.

Font window footer

There are several buttons and dropdown lists in the footer of the Font window:

	Defines zoom mode for a table of glyphs
	Decreases the zoom mode by one step
	Increases the zoom mode by one step
	Title dropdown list that defines the caption view mode for glyph cells
	Mode dropdown list that defines the current font editing mode
	Encoding dropdown list that defines the current encoding table based on the editing mode

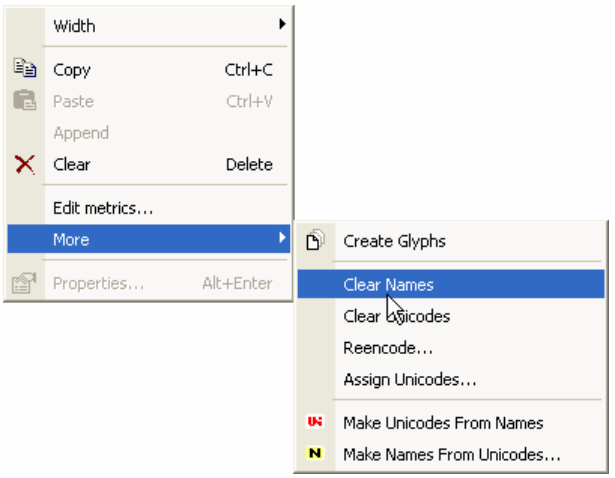
The footer of the BitFonter's window contains some additional glyph information: the current glyph name and Unicode codepoint, the total number of glyphs in the font and the number of selected glyphs:



Font window context menu

Some commands available in the Font window can be selected from the context menu.

To open the context menu, right-click anywhere in the chart area. Here is a sample of the Font window context menu:

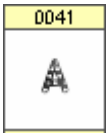


Here is what the commands mean:

Width	Allows you to easily select one of the predefined widths of the Font window
Copy	Copies the selected glyphs onto the Clipboard. Same as the **Copy** command from the **Edit** menu
Paste	Places glyphs from the Clipboard into the font starting from the first selected cell. Same as the **Paste** command from the **Edit** menu
Append	Places glyphs from the Clipboard to the cells according to their Unicode codepoints and names. Same as the **Append** command from the **Edit** menu
Delete	Deletes the selected glyphs. Same as the **Delete** command from the **Edit** menu
Edit metrics	Opens the Metrics window with the selected glyphs
More	Submenu with some useful commands from the **Glyph** menu
Properties	Opens the glyph properties panel for the current glyph.

Changing cell caption

Each cell has a caption (or title) showing the name of the glyph, its code in various forms, or its advance width.



The **Title** dropdown list

located in the Font window footer lets you select one of the caption view modes:

None	Glyph's caption will not be shown at all
Name	The glyph name
Unicode	The Unicode codepoint assigned to the glyph, in hexadecimal form
Index	The glyph index, i.e. the physical location of the glyph in the font
ASCII	The ASCII character that corresponds to the local character code
Decimal Code	The local character code in decimal form
Octal Code	The local character code in octal form
Hex Code	The local character code in hexadecimal form
Width	The glyph advance width
Left Sidebearing	The glyph's left sidebearing
Right Sidebearing	The glyph's right sidebearing
Bottom Sidebearing	The glyph's bottom sidebearing
Body width	The glyph shape width
Body height	The glyph shape height
Unicodes count	The number of Unicode codepoints assigned to the glyph.

The title view mode affects the caption display only and does not change the glyph attributes or font editing mode.

The glyphs' cells may have a gray or white background and gray or yellow caption.

A gray cell background means that there is no glyph in this place in the font. Instead of the font glyph, a sample template glyph from one of the system fonts or from the BitFonter template font is shown in the empty cell. You can switch off the templates on the **Font Window > View** page of the Tools > Options dialog box.

A gray cell caption means that the character is absent in the current encoding. Characters matching the current encoding are shown with yellow captions and reside in the upper part of the font chart. The cells with yellow captions form the so-called “yellow” zone.

You can switch off the caption by choosing *None* in the **Title** dropdown list. You can set the default title view mode and select a font for captions on the **Font Window > View** page of the Tools > Options dialog.

Font Editing Modes

BitFonter has four font window editing modes based on two major types of glyph identification — identification by names and identification by numbers (actually Unicode codepoints).

Historically, name-based encoding tables are called encoding vectors and Unicode-based encoding tables are called codepages.

The **Mode** dropdown list located in the Font window footer lets you select one of the font editing modes:

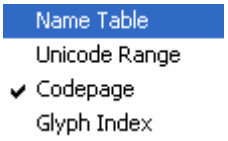
Name Table	Sets the Encoding Vector editing mode
Unicode Range	Sets the Unicode Range editing mode
Codepage	Sets the Codepage editing mode
Glyph Index	Sets the Index editing mode

The different editing modes allow you to look at the font from different points of view. But also the acts of creating new glyphs moving, and pasting glyphs work differently in different modes.

Encoding Vectors

The **Names** mode shows references between character codes and names. Encoding Vectors use one byte to encode characters; so only 256 characters of a font may be used at a time. But you may have many more characters in one font having different names and referenced by different encodings. I.e. a font may have multiple encodings, each with a different set of 256 characters.

To switch the Font window to Names editing mode, choose the **Name Table** command in the **Mode** dropdown list in the Font window footer:



To select the current encoding for the font, use the **Encoding** dropdown list to the right. When you open it you will see the big list of available encodings. Here are some of them:

Adobe Standard Encoding	Adobe standard encoding for creating Roman PostScript fonts
Adobe Symbol Font	Standard encoding for supporting fonts that include math and other characters
Default Encoding	The default encoding for all the applications in the FontLab family. This provides the best support for the Default character set on Windows
ISO 8859-1 Latin 1	ISO Latin 1 standard encoding
MacOS Roman	Standard Latin 1 encoding for Macintosh computers
MacOS Cyrillic	Standard encoding for supporting Cyrillic characters on Macintosh computers
MS Windows 1252	Standard Latin 1 encoding used in Windows

When you change the encoding you will see that the characters in the Font window are rearranged. Some characters will move below the “yellow” zone. This means that these characters (actually their names) are not covered by the selected encoding vector.

Unicode Ranges

In Unicode the standard character space is divided into *planes (*defined by the first byte of the three-byte codes*)*. Each plane (codepoints 0 — 65,535) is divided into *ranges*. Each range typically covers characters that belong to one alphabet or have common properties, like the Cyrillic range or the Hebrew or the Extended Latin.

Ranges may be of various lengths — from a few characters to several thousand characters (in the case of Kanji characters).

In BitFonter you can select any Unicode range and view your font as organized by the range. All characters with Unicode codepoints in the selected range will be arranged in order in the yellow zone at the top of the Font window.

In order to simplify working with Unicode ranges in BitFonter all the "official" ranges in the Unicode standard are subdivided into subranges. You can work with the whole range or select one of the subranges. For example, you can select the whole Cyrillic range that includes all currently used and historic Cyrillic characters, or you can select just the historic letters or only the Russian alphabet.

To switch the Font window to Unicode Range editing mode, choose the **Unicode Range** command in the **Mode** dropdown list in the Font window footer.

To select the current range or subrange, use the **Encoding** dropdown list to the right. Ranges' names are aligned to the left of the list box and the subranges' names are indented to the right. When you select the range or subrange that you want to work with, you will see the Font window change so that only glyphs from the selected range are in the "yellow" area. Of course, all the glyphs must have Unicode codepoints.

Codepages

Codepages are the tables that map local character codes (one byte long) to the Unicode codepoints. These tables have 256 records, one for each possible character code. Codepages are necessary because we need to somehow encode text written in different languages in the one-byte code space.

Because the Unicode character identification standard covers most languages it is usually used as the destination information in the codepage tables. Here is an example of fragments from two different codepages that map the same codes to different Unicode codepoints:

MS Windows 1252 Latin 1		MS Windows 1251 Cyrillic	
0xC0	0x00C0	0xC0	0x0410
0xC1	0x00C1	0xC1	0x0411
0xC2	0x00C2	0xC2	0x0412
0xC3	0x00C3	0xC3	0x0413
0xC4	0x00C4	0xC4	0x0414
0xC5	0x00C5	0xC5	0x0415
0xC6	0x00C6	0xC6	0x0416
0xC7	0x00C7	0xC7	0x0417
0xC8	0x00C8	0xC8	0x0418
0xC9	0x00C9	0xC9	0x0419
0xCA	0x00CA	0xCA	0x041A
0xCB	0x00CB	0xCB	0x041B

Many different codepages have been defined for many languages and different operating systems. BitFonter includes descriptions for more than 200 codepages - almost all the known Windows, OS/2, MS DOS, Mac OS codepages and many additional, like Russian KOI-8 and NeXT Step codepages.

In BitFonter a codepage is a filter through which you can look at your font from the environment point of view. You may include many Unicode characters in your font and see how it would work, for example, if it were installed in OS/2 with the Arabic language selected. This gives you the opportunity to easily create fonts that are properly encoded and will always work correctly.

To switch the Font window to the Codepage editing mode, choose the **Codepage** command in the **Mode** dropdown list in the Font window footer.

To change a codepage in the Font window, use the **Encoding** dropdown list to the right. You will see the Font window change. All the characters that are in the codepage will appear with yellow captions. All other characters have gray captions and reside below. Select the MS Windows 1252 Western codepage, for example, and you will see how your font will look in the Windows standard codepage.

- Note: The Codepage editing mode is the default in BitFonter. You can change the default editing mode on the **Font Window > View** page of the Tools > Options dialog.
- Note: The codepage selected in the **Encoding** dropdown list does not affect the codepage that is written to the font. Be sure the right codepage is selected in **Font Info > Encoding settings** before saving the project or individual font.

How to Make a Codepage Definition File

Codepage definition files (extension CPG) are text files that have the following structure:

```
%%FONTLAB CODEPAGE: 0xff; MacOS Cyrillic
% source:
http://www.unicode.org/Public/MAPPINGS/VENDORS/APPLE/CYRILLIC.TXT
%!Macintosh Script:7
%!Macintosh Languages:32,42,43,44,45,46,47,48,49,53,54,55,56,58,135
0x00 0x0000
0x01 0x0001
0x02 0x0002
0x03 0x0003
```

The first line of this file is an identification line that is used to set the codepage name and tell BitFonter that this file is a properly composed codepage definition file. This line must be started by the text:

%%FONTLAB CODEPAGE: 0xFFFF;

The name of the codepage follows.

All other strings starting with '%' are comments and are not interpreted by BitFonter.

The following strings are formed as pairs of two integer numbers in decimal or hex (starting with "0x") form. The first number is the local code of the character and should be in the 0-255 range. The second number is the Unicode codepoint of the character and should be in the 0-65535 (0-FFFFh) range. The special Unicode codepoint 0xFFFF is used to define codes that are not mapped to any character.

All codepages are installed in the [Shared default data folder]\Codepage\ (typically C:\Program Files\Common Files\FontLab\Codepage\ folder) and are available to all programs of the FontLab family. If you want to make a codepage file available only to BitFonter you must put the file to the [Application default data]\Codepage\ folder.

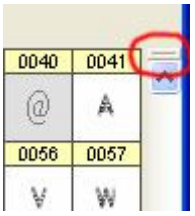
Put your custom codepage files in the [Shared user data folder]\Codepage\ folder (typically C:\Documents and Settings\Your Username\My Documents\FontLab\Shared\Codepage) if you want to make the codepage available to all recent Fontlab Ltd. applications, or in the [Application user data folder]\Codepage\ folder (typically C:\Documents and Settings\Your Username\My Documents\FontLab\BitFonter3\Codepage\) if you want to make the codepage available within BitFonter only.

Using the Font Window

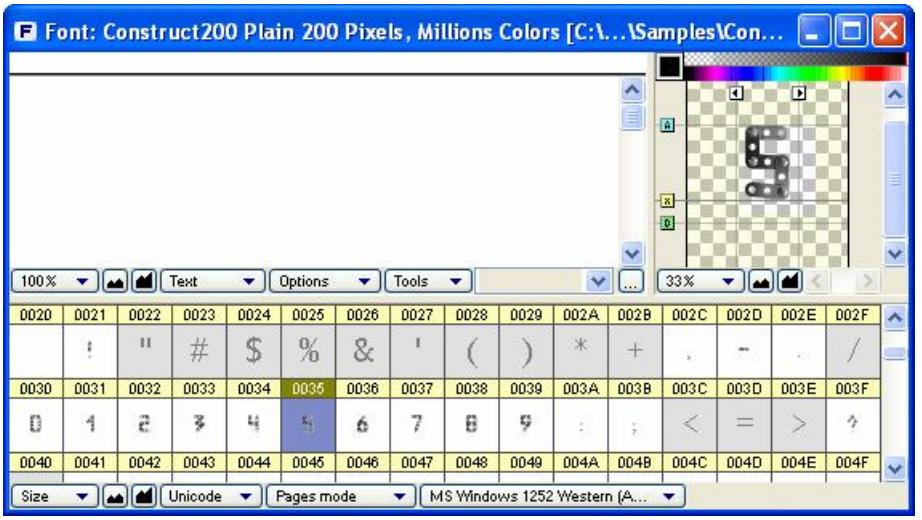
The Font window contains cells for all the font glyphs. Use the vertical scroll bar to view those glyph cells that are not visible, if any. You also may scroll with the hand tool available when you press the **Spacebar** on the keyboard.

With the Size dropdown list, you can define magnification or reduction of the Font window contents. There are 7 zoom modes available in the menu. The **Zoom-in** and **Zoom-out** buttons allow you to change cell size by one level at a time.

There is a hidden view feature in the Font window. Look for the small control above the vertical scroll bar:



Click on this small control with the mouse cursor and drag it lower along the scroll bar. Additional Font window frames will become visible:



There is a Metrics frame to the left, which is identical to the Metrics window that will be described in the ***Editing Metrics*** (on page 307) chapter. You can drag and drop glyphs there and view them while editing glyphs. You can type text in the Metrics frame with existing glyphs, but make sure you have clicked on in the frame before you start typing. You can change the zoom mode and strings of text in the Metrics frame. Double-clicking on the glyph here will open its Glyph window and let you edit the glyph.

Also the Glyph frame becomes visible above the character set table. If it is not visible, make it visible with the second small control (near the first one that you used to open the sample panel):

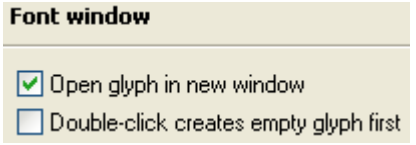
This part of the window almost completely duplicates the Glyph window that will be described in the ***Editing Glyphs*** (on page 225) chapter. If this frame is active, the drawing tools become available and you can edit an individual glyph of the font.

You can open and close Font window frames by double-clicking on the described controls.

Of course, you can resize the whole Font window by dragging its size box at the right bottom corner.

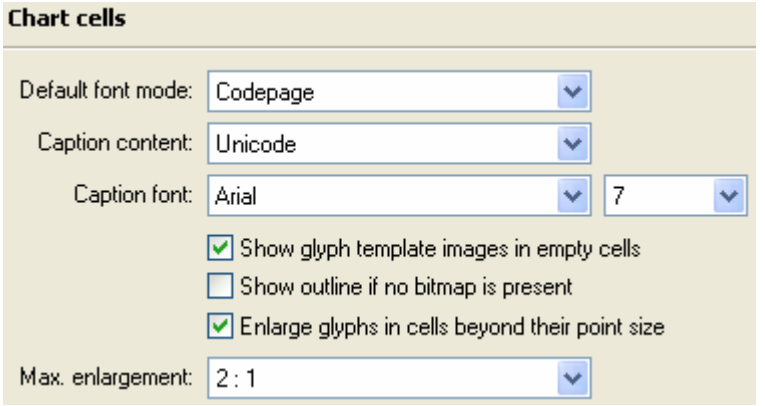
Customizing the Font Window

To customize the view in the Font window, select the **Tools > Options** command in the **Tools** menu and refer to **Font Window** pages of the Tools > Options dialog:



The **Open glyph in new window** option, when checked, tells BitFonter to open a double-clicked on glyph in a new Glyph window. If this option is off (it is on by default), all glyphs are opened in the same window for editing. (I.e. selecting a new glyph for editing will cause the glyph in the window to be replaced). Pressing the **Ctrl** key when double-clicking on the glyph reverts the meaning of this option.

When **Double-click on creates empty glyph first** is checked double-clicking on an empty cell in the Font window creates the glyph but does not open it. To open the glyph for editing you need to double-click on it again. By default this option is off and double-clicking on an empty cell once creates and opens the glyph.



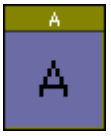
Define the default font editing mode with the help of the **Default font mode** dropdown list.

Refer to the ***Font Window Options*** (on page 48) section for more information.

Selecting Glyphs

You may need to select glyphs in the Font window for various purposes. Selected glyphs can be copied, deleted, moved, transformed etc.

To select a single glyph click on its cell with the mouse cursor. The selected cell will be shown highlighted:



To select more than one glyph, click on every cell you want to select while holding down the **Shift** key or hold down the left mouse button on the first or last cell of your selection and drag the cursor across the cells you want to select. You will see the selection highlighted. If you drag the cursor outside the visible part of the cell table, it will scroll accordingly.

You can also use the keyboard to select cells in the Font window:

Arrow keys	Select the neighbor cell of the currently selected cell
Ctrl+Arrow	Select the neighbor of the neighbor cell of the currently selected cell
End	Selects the rightmost cell on the current row
Home	Selects the leftmost cell on the current row
Page Up and Page Down	Selects the cell one screen up or down relatively to the currently selected
Ctrl+Home	Selects the first cell of the table
Ctrl+End	Selects the last cell of the table
Alphanumeric keys	When pressed separately, select the characters assigned to these keys by the current keyboard layout. When typed quickly one by one, select cells by the typed name, if available

You can use keyboard keys for selecting more than one cell by holding down the **Shift** or **Ctrl** key.

To select all the glyphs of the font, use the **Select All** command in the **Edit** menu.

To cancel your selection, click the left mouse button in the area between the rightmost cell and the scroll bar or choose the **Deselect** command in the **Edit** menu.

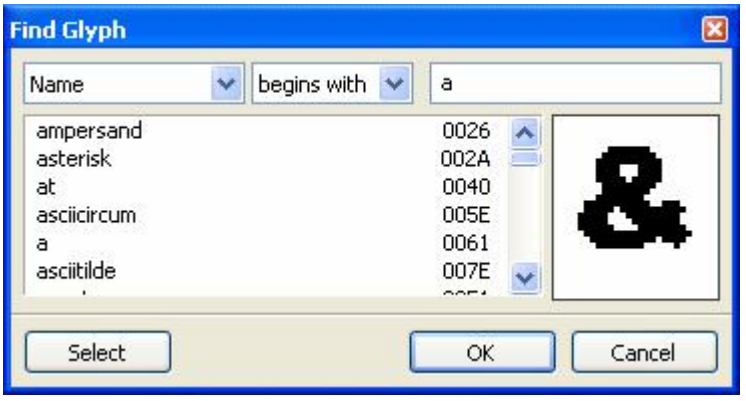
You can use the **Select Glyphs** hierarchical menu in the **Edit** menu to select cells by their attributes:

Encoded Symbols	Selects all the cells with glyphs having Unicode codepoints represented in the font's codepage
Unencoded Symbols	Selects all glyphs having Unicode codepoints not represented in the font's codepage
Named Symbols	Selects all cells with glyphs having names
Unnamed Symbols	Selects all cells with glyphs not having names
Symbols with Unicodes	Selects all cells with glyphs having a Unicode codepoint
Symbols without Unicodes	Selects all cells with glyphs not having a Unicode codepoint

Please note that the range of encoded glyphs does not depend on the editing mode or the **Encoding** dropdown list selection. Encoded glyphs are those with Unicode codepoints represented in the codepage that is set in the **Font Info > Encoding settings** dialog.

Searching for Glyphs

Sometimes you need to find a particular glyph in your font. Select the **Find Glyph** command in the **Edit** menu. You will see a dialog:



To find a glyph:

1. In the top dropdown lists select the parameter by which you want to search for the glyph(s). You can search for glyphs by name, Unicode codepoint, advance width, height, or horizontal or vertical offsets.
2. In the edit box to the right of the menus enter the criteria (depending on your selection) that will be used to find the glyph.
3. The names of all glyphs that match the entered data will appear in the list. Click on the **Select** button and all the glyphs will be selected in the Font window or select the glyph from the list that you want (its preview will appear in the preview panel) and click on the **OK** button.

Creating New Glyphs

To create a new glyph in an open font, select one or more empty cells (gray background) and select **Create Glyphs** in the **Glyph** menu. The cell background will become white, which means you have created the glyph definition.

When you double-click on an empty cell BitFonter creates the glyph and opens it in the Glyph window. Switch on the **Double-click on creates empty glyph first** option on the **Font Window** page of the Tools > Options dialog box to not open created glyphs.

The created glyph gets the appropriate name and Unicode codepoint from the encoding selected in the **Encoding** dropdown list.

Moving Glyphs

You can change the position of glyphs in the Font window and move them to a new place.

To move a glyph in the font table:

1. Select the glyphs that you want to move.
2. Position the mouse cursor on the selected glyphs.
3. Hold down the left mouse button and drag the selection to the new position.
4. Release the left mouse button to finish moving.

If you move glyphs over existing glyphs, the existing glyphs will be saved at the end of the table. If you do not want to save existing glyphs, uncheck the **Keep replaced glyphs under new names** check box in the **Font Window > Drag & Drop** page of the Tools > Options dialog. The existing glyphs will be overwritten.

When moving glyphs their names and Unicode codepoints change according to their position in the current encoding or codepage. When glyphs are moved in the Index mode they do not replace other glyphs but just change their position in the font file.

Copying and Pasting Glyphs

To copy selected glyphs, select the **Copy** command from the **Edit** menu. The selected glyphs will be placed on the Clipboard and can be pasted into the same font or into another font by the **Paste** command.

To paste copied glyphs over existing glyphs, select the cell where you want the pasting to start and select the **Paste** command in the **Edit** menu. Glyphs from the Clipboard will replace the glyphs starting with the selected cell and the replaced glyphs will be saved at the end of the font table. If you do not want to save the replaced glyphs, uncheck the **Keep replaced glyphs under new names** check box in the **Font Window > Drag & Drop** page of the Tools > Options dialog. The replaced glyphs will be overwritten.

If you select the **Cut** command instead of the **Copy** command, the glyphs will be copied to the Clipboard but will be deleted from the source positions.

do not forget that you can use the **Undo** command if you want to undo any of these operations.

When pasting glyphs their names and Unicode codepoints change according to their position in the current encoding or codepage.

If copied glyphs have outlines in their outline layer, the outlines are pasted together with the bitmaps. To paste copied outlines without bitmaps, use the **Paste Outline** and **Append Outline** commands. To learn more about glyph outlines, refer to the ***Outline Operations*** (on page 300) section.

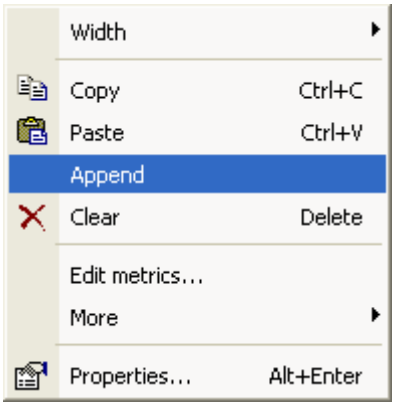
Appending Glyphs to the Font

Instead of the **Paste** command you can use the **Append** command from the **Edit** menu or the Font window context menu to add glyphs from the Clipboard to the font.

When BitFonter appends glyphs, it respects the glyph names and Unicode codepoints, so on the first attempt glyphs will be placed in the expected code positions in the font.

Here is an example. Your first font contains Latin glyphs but has no Cyrillic glyphs. A second font is a Cyrillic font with the matching style and you want to add Cyrillic support to the first font.

1. Select the Cyrillic glyphs in the second font (this will be easy if you select the Mac OS Cyrillic codepage or the Cyrillic Unicode range) and copy them to the Clipboard.
2. Return to the first font; right-click on the Font window; and click on the **Append** command in the context menu:



3. The Cyrillic glyphs will be appended to the font with their correct Unicode codepoints and names, so you will not have to re-map the font.

Deleting Glyphs

To remove glyphs from the font:

1. Select the glyphs that you want to remove.
2. Select the **Delete** command from the **Edit** menu or press the **DEL** key on the keyboard.

The deletion can be undone by using the **Undo** command in the **Edit** menu.

Transforming Glyphs

Sometimes you need to transform a set of glyphs or the whole font. The most efficient way to do this is to use the transformation commands from the **Tools** menu.

➘ Note: Some commands are not available in the **Tools** menu if you are editing a black & white font. The transformation commands are described in the ***Operations*** (on page 270) section of the ***Editing Glyphs*** (on page 225) chapter.

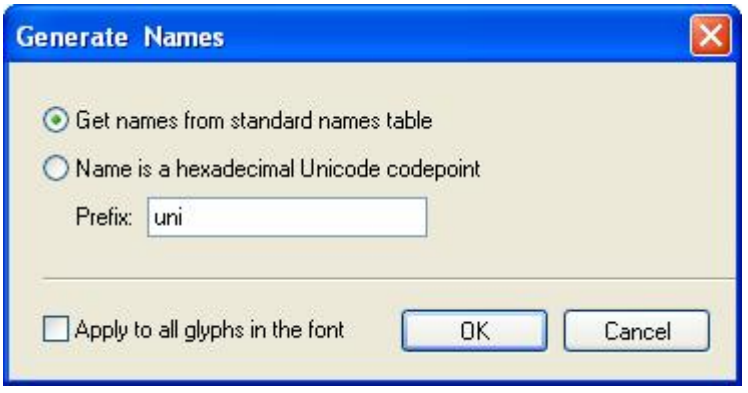
To transform a set of glyphs, select glyphs and choose the appropriate command in the **Tools** menu. If no cells were selected the operation will be applied to all glyphs in the font.

Renaming and Reencoding Glyphs

Usually it is not necessary to manually rename or reencode glyphs because their names, Unicode codepoints and codes are assigned automatically when you create glyphs in the Font window. If you want to see the information for each separate glyph and correct it, you can use the Glyph Properties window described in the ***Editing Glyph Properties*** (on page 234) section of the ***Editing Glyphs*** (on page 225) chapter. But if you want to work with a set of glyphs or the whole font, the commands from the **Glyph > Name and Unicode** menu become useful.

To clear names or Unicode codepoints of glyphs, use the **Clear Names** and the **Clear Unicode** commands from the **Glyph > Name and Unicode** menu.

To name glyphs by their Unicode codepoints, select the **Generate Names** command in the **Glyph > Name and Unicode** menu. The Generate Names dialog will appear:



There are two ways of naming glyphs. You may choose that names be generated from Unicode numbers or that they be retrieved from the table of standard names.

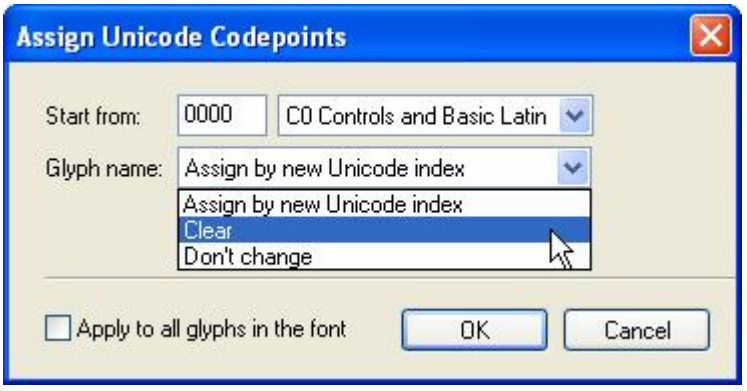
Select the **Name is a hexadecimal Unicode codepoint** option for the first way. Enter a prefix for the glyph name, for example, '0x' and click on **OK**. The glyphs will get names identical to their Unicode codepoints with a '0x' prefix. For example, the glyph “@”, having a Unicode codepoint '0040', will get the name “0x0040”.

Select the **Get names from standard names table** option to generate standard glyph' names. BitFonter will find the name for each glyph by its Unicode codepoint in the “STANDARD.nam” file located in the [Shared data]\Mapping\ folder.

To generate Unicode codepoints from glyph names, select the **Generate Unicode** command in the **Glyph > Name and Unicode** menu. As with some commands from the **Tools** menu, a dialog appears if no glyphs were selected. Click on **Yes** to process the whole font. The glyphs will have their Unicode codepoints generated from their names automatically. For example, the character “@” having name “uni0040” will get the Unicode codepoint '0040' and the character with the name “at” will also get Unicode codepoint '0040'.

To assign new Unicode codepoints to glyphs:

1. Select the **Assign Unicode** command in the **Glyph > Name and Unicode** menu. The following dialog appears:



2. Set the starting Unicode codepoint in the **Start from** field by typing it in. You can also set the starting Unicode codepoint by selecting the appropriate Unicode range in the **Ranges** dropdown list.

3. Select the method for assigning glyph names in the **Glyph name** dropdown list. Set *Assign by new Unicode index* item to rename glyphs according to new codepoints, or set *Clear* to remove glyph names, or set D*o not Change* to leave glyph names unchanged.
4. Set the **Apply to all glyphs in the font** option to assign new Unicode codepoints to all the glyphs in a font. If this option is unchecked, only the selected range of glyphs will be reindexed.
5. Click on the **OK** button to assign new codepoints to the glyphs.

➘ Note: Unicode codepoints are assigned to glyphs by their current order in the Font window.

Reencoding Glyphs Automatically

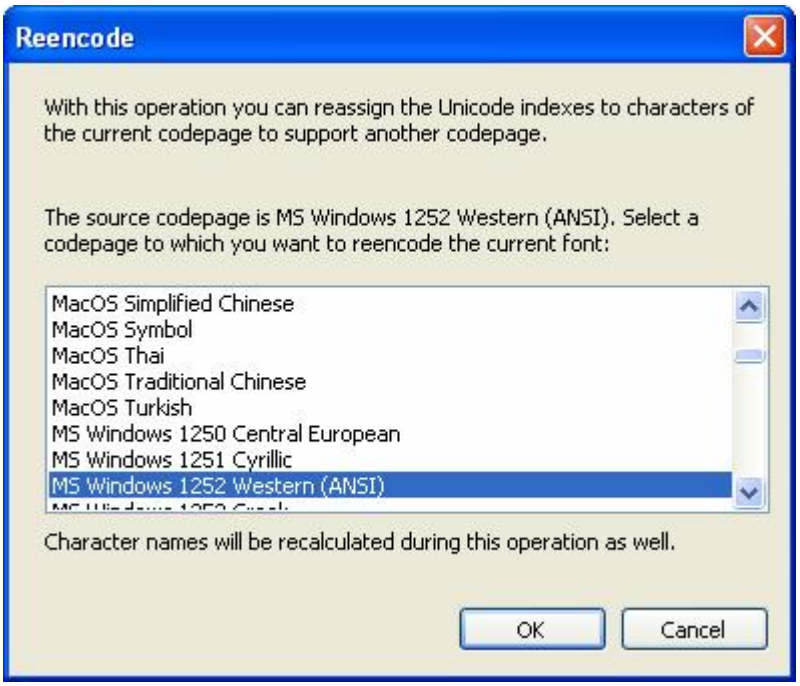
Sometimes, for example when you get a font with improper encoding, you may need to reencode it quickly without performing multiple dragging or renaming operations. This is where you need the automatic reencoding feature of BitFonter.

There are three different ways of reencoding a font: two ways when it is open in a Font window and one way to reencode all the fonts in the font family.

Reencoding a Font

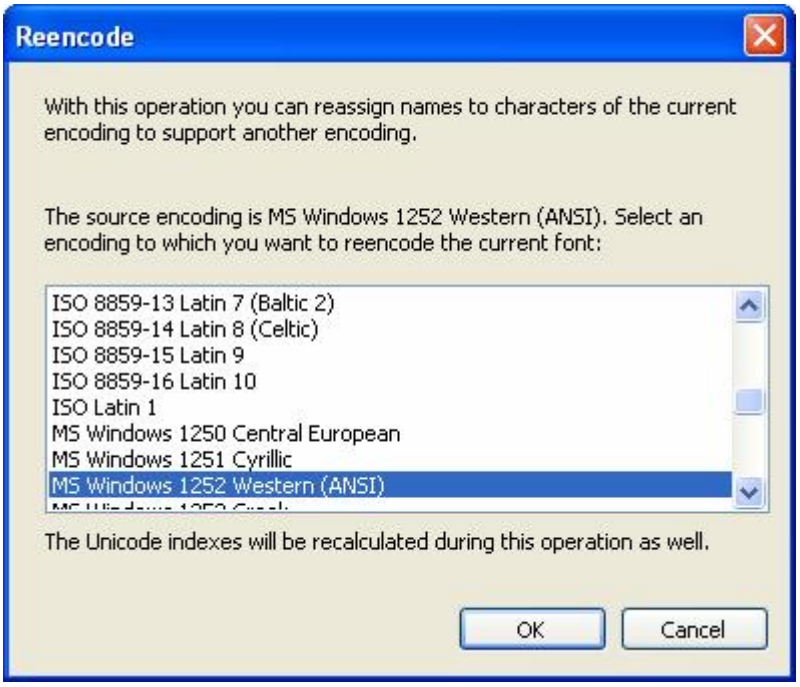
Reencoding a font while it is open in its Font window can be performed in two ways: by Codepage or by Name Table depending on the current editing mode.

To reencode glyphs by Codepage, switch the Font window to the **Codepage** editing mode using the **Mode** dropdown list at the bottom of the Font window and then select the **Reencode** command from the **Glyph > Name and Unicode** menu. The Reencode dialog appears:


Reencode
With this operation you can reassign the Unicode indexes to characters of the current codepage to support another codepage.
The source codepage is MS Windows 1252 Western (ANSI). Select a codepage to which you want to reencode the current font:
MacOS Simplified Chinese
MacOS Symbol
MacOS Thai
MacOS Traditional Chinese
MacOS Turkish
MS Windows 1250 Central European
MS Windows 1251 Cyrillic
MS Windows 1252 Western (ANSI)
Character names will be recalculated during this operation as well.
OK
Cancel

You can see the list of all the available codepages in the dialog. The text above the list contains the source codepage of your font. To start reencoding, select the destination codepage from the list and click on the **OK** button. The reencoding algorithm in this case is as follows: BitFonter gets each glyph's Unicode codepoint and changes it to the corresponding codepoint from the destination codepage, then BitFonter changes the name of each glyph to the name of its new Unicode codepoint by looking it up in the "STANDARD.NAM" file.

To reencode glyphs by Name Table, switch the window to the **Names** editing mode using the **Mode** dropdown list at the bottom of the Font window and then select the **Reencode** command from the **Glyph > Name and Unicode** menu. The Reencode dialog appears:


Reencode
With this operation you can reassign names to characters of the current encoding to support another encoding.
The source encoding is MS Windows 1252 Western (ANSI). Select an encoding to which you want to reencode the current font:
ISO 8859-13 Latin 7 (Baltic 2)
ISO 8859-14 Latin 8 (Celtic)
ISO 8859-15 Latin 9
ISO 8859-16 Latin 10
ISO Latin 1
MS Windows 1250 Central European
MS Windows 1251 Cyrillic
MS Windows 1252 Western (ANSI)
The Unicode indexes will be recalculated during this operation as well.
OK
Cancel

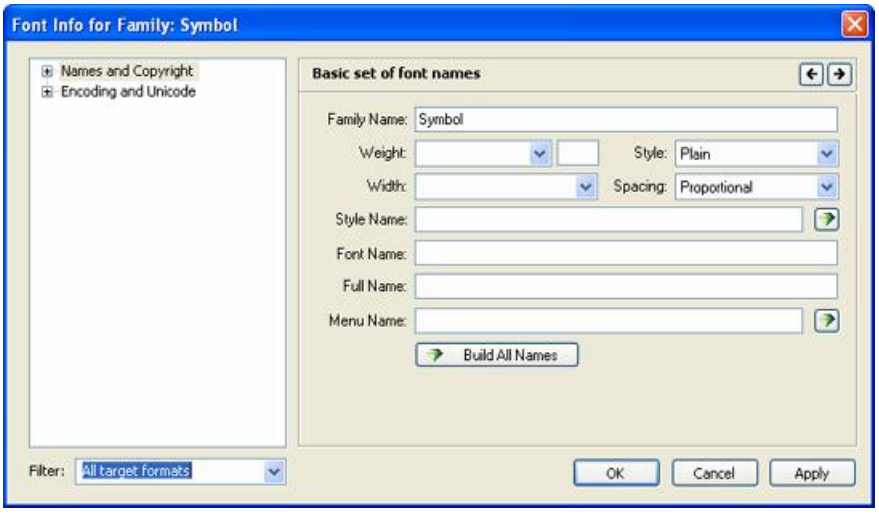
You will see the list of all the available Encoding Vectors in the dialog. The text above the list contains the name of the source encoding of your font. To start reencoding, select the destination encoding from the list and click on the **OK** button. The reencoding algorithm in this case is as follows: BitFonter gets each glyph's name and changes it to the corresponding name from the destination encoding, then BitFonter changes the Unicode codepoints of glyphs accordingly to the "STANDARD.nam" file.

Reencoding a Font Family

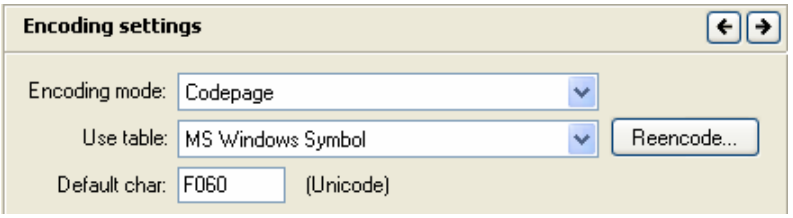
Reencoding a font family means reencoding all the bitmap fonts in the family with a few mouse clicks. Only the Codepage algorithm can reencode a family of fonts. Note that reencoding only one font in a family containing multiple fonts is useless.

To reencode the whole font family (e.g. Windows font set):

1. Select the family name (or icon) in the Project window and choose the **Font Info** command from the **File** menu. The Family Info dialog appears:



2. Select the **Encoding and Unicode** page to see the encoding settings:

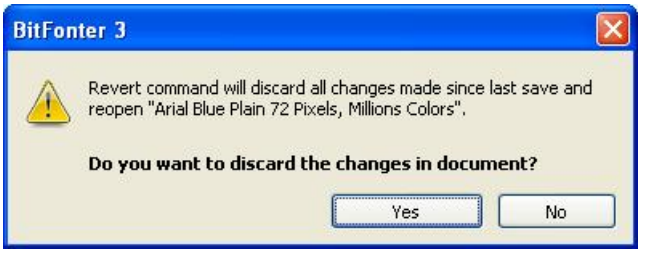


3. In the **Use table** dropdown list you see the current codepage of your font family. Leave it unchanged or select another one - this will be the source codepage during reencoding.
4. Click on the **Reencode** button to the right of the **Use table** dropdown list . The Reencode dialog with the list of all available codepages appears.
5. To start reencoding, select the destination codepage from the list and click on the **OK** button. The reencoding algorithm in this case is the same as when you reencode a single font by codepages.

➘ Note: It is important to note that BitFonter does nothing with outline fonts that might be the part of the font family. The names and/or Unicode codepoints in these fonts would not change.

Reverting a Font

Anytime you are working with an open font you can revert the font to the last saved version. To do this, select the **Revert** command in the **File** menu. You will see the revert confirmation dialog:


BitFonter 3
Revert command will discard all changes made since last save and reopen "Arial Blue Plain 72 Pixels, Millions Colors".
Do you want to discard the changes in document?
Yes
No

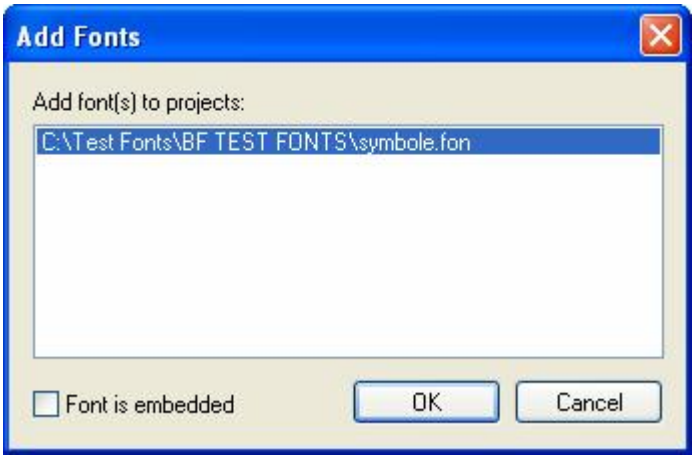
Click on the **Yes** button if you really want to discard all changes to the font since it was last saved. The last saved version of the font will be read from the disk in this case. Click on the **No** button if you change your mind.

Adding a Font to the Project

You may want to combine similar or different fonts in projects for various reasons.

To add a font to the project:

1. Open the project in a Project window.
2. Open the font that you want to add in the Font window and select the **Add to Project** command in the **Font** menu. A dialog asking your confirmation and showing a list of open projects will appear:



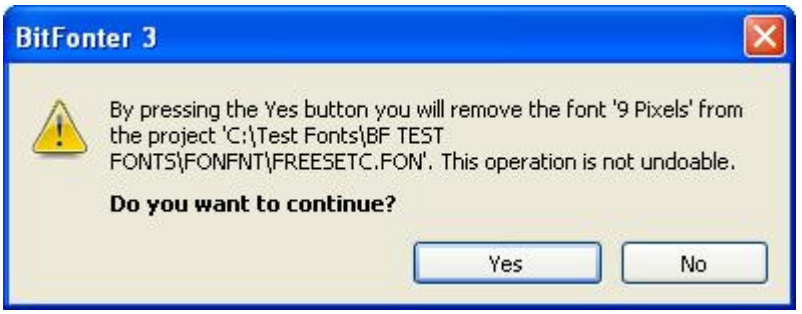
3. Select the project name in the list and click on the **OK** button to add to the project, or **Cancel** to abort. To add a font with the OpenType SBIT embedding mark set, check the **Font is embedded** option. This will allow you to add a bitmap font with more than 255 glyphs.

If the **Add to Project** menu command is not available it means that there is no open project or the font is in the project already.

Removing a Font from the Project

You may need to remove one or more fonts from the project. There are two ways of removing fonts:

1. If the Font window with the font you are removing is open, select the **Remove from Project** command in the **Font** menu. You will see the confirmation alert:


BitFonter 3
By pressing the Yes button you will remove the font '9 Pixels' from the project 'C:\Test Fonts\BF TEST FONTS\FONFNT\FREESETC.FON'. This operation is not undoable.
Do you want to continue?
Yes
No

   To confirm deletion, click on the **Yes** button, or press **No** to cancel the operation. When you confirm deletion the font will be deleted from the project, but remain open in its Font window. You may choose to save it, add it to another project or close it without saving to disk.

2. To remove a font or fonts that are not open, select them in the Project window and use the **Cut** or **Delete** command in the **Edit** menu or press the **DEL** key on the keyboard.

Important Notes:

- Remember that removing the font from the native BitFonter format project does not affect the real font file that resides separately on the disk along with the project file. Such a deletion removes a reference to that file only. To dispose of the font forever, move it to the Recycle Bin.

- But on the other hand, when you are working with a Windows font set, or other font of the project type, removing the font from the project *will* delete the real font entry from the composite file. So be very careful when deleting fonts from projects.

Duplicating a Font

It may sometimes be useful to duplicate an open font. You already know how to duplicate fonts that are part of a project. But when the font you are editing does not belong to any project you may also duplicate it.

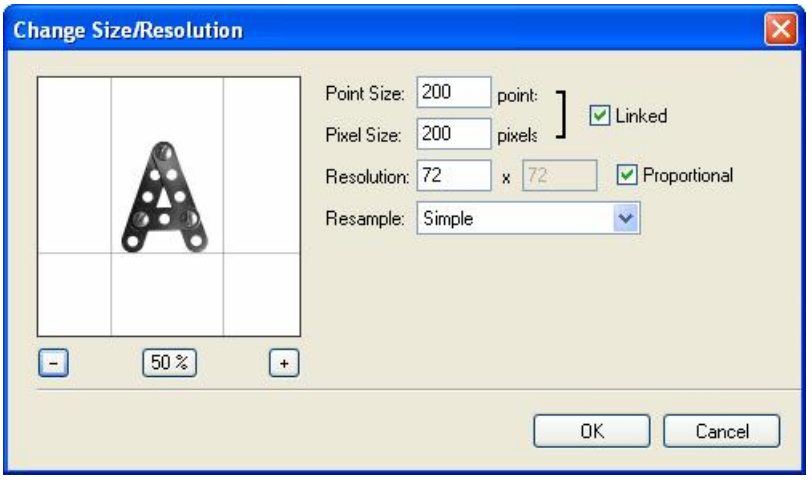
To make a copy of an open font, select the **Duplicate** command from the **Font** menu. The current font will be duplicated in memory and shown in a separate Font window. You may edit it in any way and save it. This technique allows you to easily create differently customized versions of one original font.

Be careful using the two different **Duplicate** commands from the **Edit** and the **Font** menus. The **Duplicate** command from the **Edit** menu duplicates fonts right in the project, placing them into the project. You need not save such duplicates separately. The **Duplicate** command from the **Font** menu duplicates an open font in memory only. In this case you will need to save it later in an appropriate format.

Changing Font Size and/or Resolution

Sometimes you may need to change the point size of your font or even the font resolution. You can do this in two ways. The first way is to change the size of all the glyphs in the font with the help of the **Scale** operation (see the ***Operations*** (on page 270) section in the ***Editing Glyphs*** (on page 225) chapter) and then change the font information in the Font Info dialog (see the ***Editing Font Properties*** (on page 90) section). But this is not the best and the easiest way. it is better to use the second way: the special command in the **Font** menu.

Once you decide to change the font size or resolution, open the font and select the **Size/Resolution** command in the **Font** menu. The Change Size/Resolution dialog will appear:



Enter the new size in points or in pixels. If the **Link** option is checked, the Pixel Size will be updated while you enter the Point Size and vice versa. If the **Link** option is not checked, changing the Point or Pixel sizes will change the font resolution in the corresponding fields. You can enter new font horizontal and vertical resolutions. Check the **Proportional** option to keep horizontal and vertical resolution equal. When the vertical font resolution changes the Pixel Size also changes while the Point Size of the font remains unchanged.

There are several predefined methods for scaling font glyphs included in the **Resample** dropdown list. If any choice other than the *Simple* resampling method is selected, you can tune it up with the slider to get the best result for your particular font. Note that what is good for one font will not necessarily be good for another.

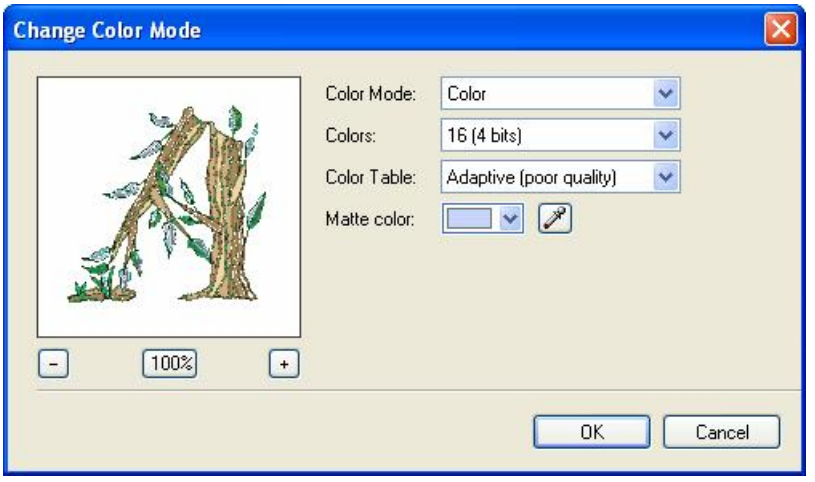
Look at the preview in the preview field to the left to keep track of changes. Use the zoom buttons to reduce or enlarge the preview. You can view the original glyph shape by pressing and holding down the **Ctrl** key on the keyboard.

When finished, click on the **OK** button to scale your font, or click on **Cancel** to abort. When changing font size and resolution BitFonter automatically recalculates font metric information.

➘ Note: If the scaled font contains glyphs with an outline layer, then the outlines are scaled proportionally.

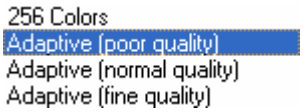
Changing Font Color Mode

While the Font window is open you can change the font color mode, i.e. the number of available colors. Select the **Color Mode** command in the **Font** menu and you will see the Change Color Mode dialog:



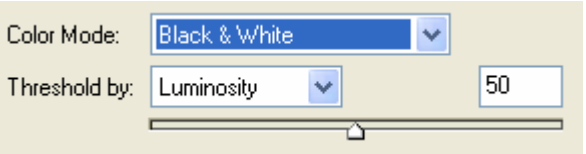
Select a new **Color Mode**, number of **Colors** and **Color Table** for the font you are editing in the corresponding dropdown lists.

You may want BitFonter to generate a custom color table for your font. This is possible in 4, 16 and 256 color modes only when you change the number of colors in the font. Select one of the available *Adaptive* color tables:



Select the **Matte color** in the dropdown list or with the help of the **Eye Dropper** tool if needed.

If you are going to make a black & white font from a color or grayscale font, you have to define the threshold parameter for conversion:



See the description of this parameter in the ***Threshold*** (on page 282) section in the ***Editing Glyphs*** (on page 225) chapter.

You can view the preview of the glyph in the preview field of the dialog box. To see the original glyph sample, simply press and hold down the **Ctrl** key.

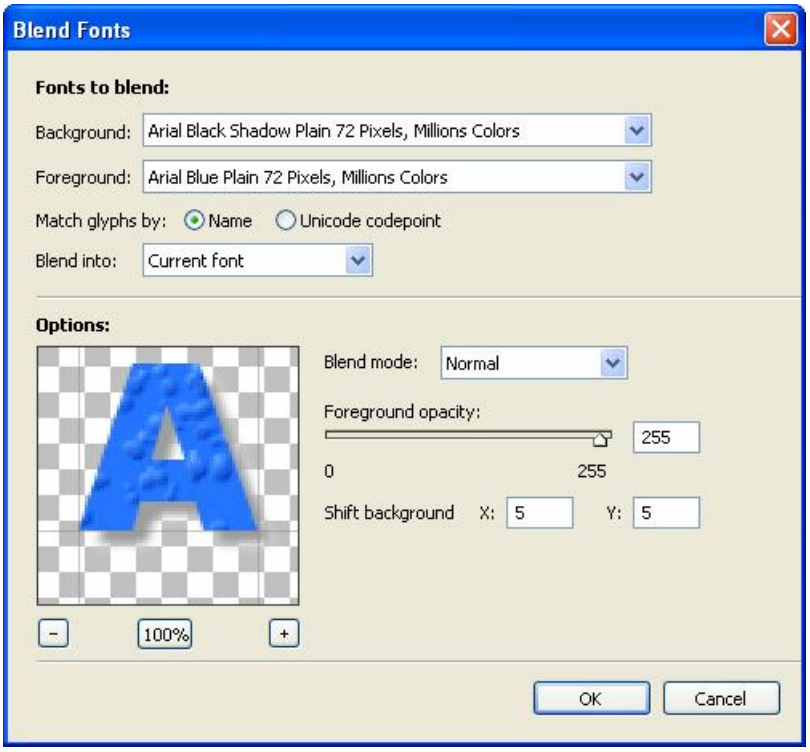
Click on the **OK** button to finish the operation, or **Cancel** to leave the font color mode unchanged.

You can undo this operation using the **Edit > Undo** command.

Blending Fonts

By blending fonts we mean the special operation on two fonts when each glyph from the “foreground” font overlaps the corresponding glyph of the “background” font, forming a new glyph. Combined with other operations (see the ***Operations*** (on page 270) section in the ***Editing Glyphs*** (on page 225) chapter) this special operation adds tremendous possibilities to PhotoFont editing techniques.

To combine the glyphs of two open fonts, select the **Blend** command in the **Font** menu. The Blend Fonts dialog will appear:



You will see two groups of options in the dialog.

Fonts to Blend

Select the two fonts that you want to blend here. Select the background font in the first dropdown list and the foreground font in the second dropdown list. The menus contain the names of open fonts only. The glyphs from the foreground font will overlap (i.e. obscure) the glyphs from the background font.

The glyphs can be matched by their names or Unicode codepoints. Select the corresponding attribute for matching glyphs in the two fonts:

you will get the best result if the two fonts have identical sets of glyphs with correlating attributes.

Select the destination where the font resulting from the blender will be placed. You may choose either the current font or the new font. If you select the *Current Font* item, then the font that had been the current before you selected the **Blend** command will be the destination font. If you select the *New Font* item, then a new font in a new window will be created by the blending procedure:

New font will always have millions of colors regardless of the source fonts' color mode.

➘ Note: If source fonts contain glyphs with an outline layer, glyph contours are not blended. Only the outlines of the foreground font are placed into the destination font.

Blend Options

Use the **Blend Mode** dropdown list to select the foreground and background color blending mode. The following modes are available in the **Blend Mode** dropdown list:

Normal	The default mode. The colors of background and foreground pixels are not blended if no transparency is applied (for 1-16 bit fonts). The colors of background and foreground pixels are blended according to their transparency values if 32-bit fonts are blended
Behind	Simply moves the background pixels to the foreground. As a result the foreground glyph goes to the background
Multiply	Looks at the color information and multiplies the background color by the foreground color. The result color is always a darker color. Multiplying any color with black produces black. Multiplying any color with white leaves the color unchanged
Screen	Looks at the color information and multiplies the inverse of the background and foreground colors. The result color is always a lighter color. Screening with black leaves the color unchanged. Screening with white produces white
Soft Light	Darkens or lightens the colors, depending on the background color. The effect is similar to shining a diffused spotlight on the image. If the background color (light source) is lighter than 50% gray, the image is lightened, as if it were dodged. If the background color is darker than 50% gray, the image is darkened, as if it were burned
Hard Light	Multiplies or screens the colors, depending on the background color. The effect is similar to shining a harsh spotlight on the image. If the background color (light source) is lighter than 50% gray, the image is lightened, as if it were screened. If the background color is darker than 50% gray, the image is darkened, as if it were multiplied
Color Dodge	Looks at the color information and brightens the background color to reflect the foreground color. Blending with black produces no change
Color Burn	Looks at the color information and darkens the background color to reflect the foreground color. Blending with white produces no change

Darken	Looks at the color information and selects the background or foreground color - whichever is darker - as the result color. Pixels lighter than the foreground color are replaced, and pixels darker than the foreground color do not change
Lighten	Looks at the color information and selects the background or foreground color - whichever is lighter - as the result color. Pixels darker than the foreground color are replaced, and pixels lighter than the foreground color do not change
Difference	Looks at the color information and subtracts either the foreground color from the background color or the background color from the foreground color, depending on which has the greater brightness value. Blending with white inverts the background color values; blending with black produces no change
Hue	Creates a color with the luminance and saturation of the background color and the hue of the foreground color
Saturation	Creates a color with the luminance and hue of the background color and the saturation of the foreground color
Luminosity	Creates a color with the hue and saturation of the background color and the luminance of the foreground color
Copy Red	Creates a color by copying the foreground red channel to the background. Other foreground channels are not used
Copy Green	Creates a color by copying the foreground green channel to the background. Other foreground channels are not used
Copy Blue	Creates a color by copying the foreground blue channel to the background. Other foreground channels are not used
Copy Opacity	Applies the opacity of the foreground color to the background color. White is treated as 100% transparent.

Use the **Foreground Opacity** slider to define the opacity of the foreground glyphs. Move it to the left to make foreground glyphs transparent or set the value to 255 to make the foreground fully opaque.

Enter the values for horizontal and vertical shift of the background glyph shapes relative to the foreground glyph shapes. The positive values move the background right and down while the negative values move it left and up.

Blend Example

Take a look at the following example font result and the algorithm to create it:

To make this font, follow these 5 steps:

1. Select the **Open Installed** command in the **File** menu. Choose the font "Times New Roman Bold" as the source font for import in the dialog. Set the destination **font size** to 64 points, set the **color mode** to "color" and the number of **colors** to "Millions (32 bits)" and click on the **Import** button. Now you have a font named "Times New Roman Plain 64 Pixels, Millions Colors".
2. Select the **Duplicate** command in the **Font** menu to make another copy of this font in memory.
3. Select the **Gaussian Blur** command in the **Tools > Filter** menu, type "40" as the radius, check the **Apply to all glyphs in the font** option and click on **OK**. Now you have the background font. You can give it a new name in the Font Info dialog ("Shadow", for example).

4. Move to the other window with the original (imported in step 1) font and make it current. Select the **Colorize** command in the **Tools > Filter** menu. Choose *Picture* in the **Assign Mode** dropdown list and click on the **Browse** button to locate the source image. Find the "water.png" image in the *Patterns* folder in the BitFonter folder and select it as the colorizing pattern. Choose the *Whole* item in the **Alignment** dropdown list, check the **Apply to all glyphs in the font** option and click on the **OK** button. Now you have the foreground colorized font. You can give it a new name in the Font Info dialog ("Water", for example).

5. Select the **Blend** command in the **Font** menu. Select the blurred font as background and the colorized font as foreground. Enter "4" as the horizontal and vertical shift values, set foreground opacity to "255", select *Normal* in the **Blend Mode** dropdown list and *New Font* as the destination font. Click on the **OK** button when finished. Voila! There's your patterned and shadowed font in the new Font window!

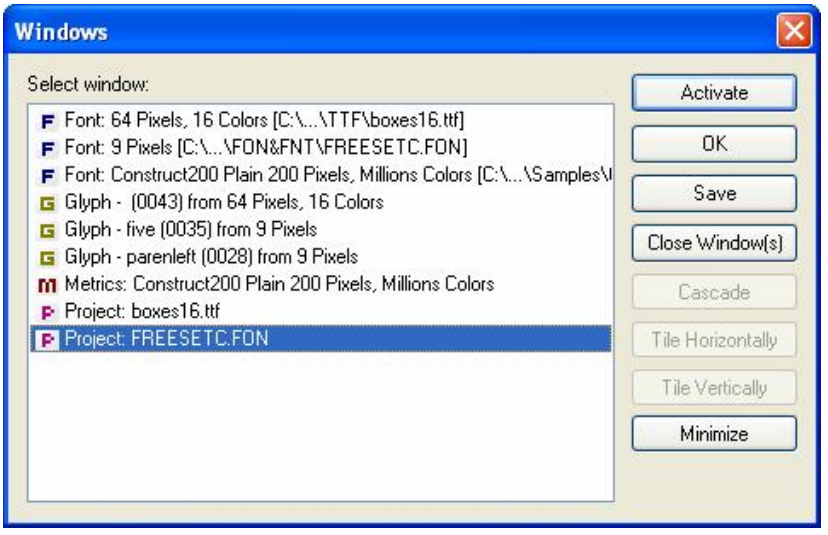
Using the Window Menu

The easiest way to manage open windows is to use the **Window** menu. It contains some very useful commands:

Cascade	Organizes open windows in a cascade like in the image above
Tile horizontally **Tile vertically**	Organizes windows like tiles on a rectangular floor
Windows...	Opens the windows management dialog box.

To open the Metrics window for the current Font window, select the **Metrics Window** command in the **Window** menu. Every open Font window may have its own Metrics window. We will describe this later in the ***Editing Metrics*** (on page 307) chapter.

Choose the **Windows** command and you will see a dialog box:



Most of the dialog box is covered by the list of open windows. Select one of the windows in the list and click on the **Activate** button to activate that window and move it to the top.

To close one or more windows, select them in the list and click on the **Close Window(s)** button.

Select two or more windows in the list and click on **Cascade**, **Tile Horizontally** or **Tile Vertically** to perform one of the operations only with the selected windows. All other windows will be automatically minimized.

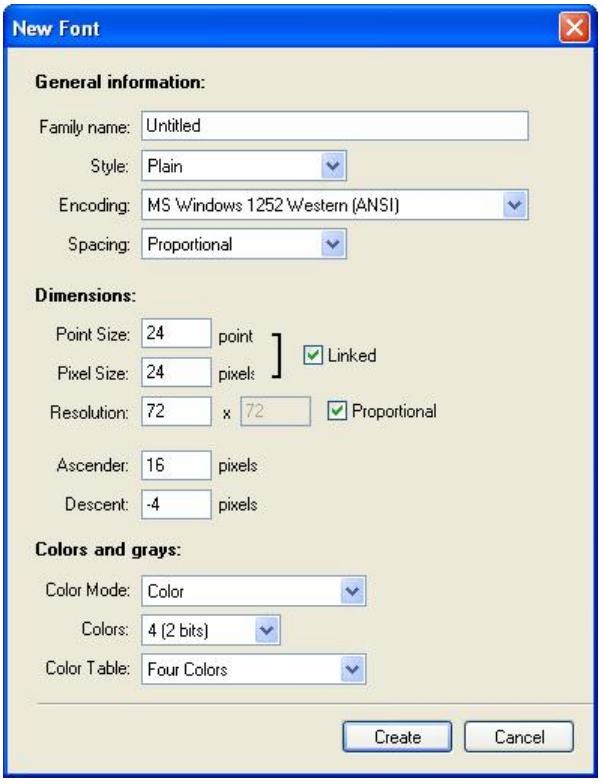
Use the **Minimize** button to minimize selected windows.

Creating and Saving Fonts

Proper font creation and saving (as well as importing and exporting) are important stages of the font design process. Understanding these stages is half of success in making a font. This chapter will show you how to create a new font from the very beginning; import glyphs from other sources; and generate a good final font product.

Creating a New Font

To create a new empty font (without any glyphs), select the **Font** command from the **File > New** menu or click on the button on the Standard toolbar if available. Another way is to select the **Create New Project Font** command in the **Project** menu while the project window is open. The New Font dialog will be displayed:



Here you define the main characteristics of the font that you will create.

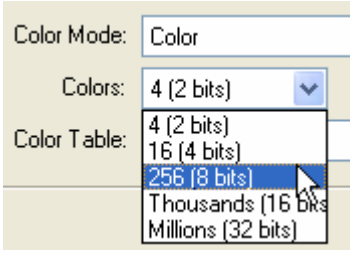
Give your font a name by typing it in the **Family Name** text box. Select the style for your font in the **Style** dropdown list. The **Spacing** dropdown list defines the type of font from the point of view of the glyph advance width. Select *Proportional* spacing if you are going to create a common font with varying glyph widths, or *Monospaced* spacing, for a font with equal glyph widths, like the 'Courier' font. Select the codepage in the **Encoding** dropdown list for the proper template in the Font window. You can change the codepage anytime later.

Define the font size and resolution. Enter the size in points or in pixels as you wish. If the **Linked** option is checked, the **Pixel Size** will be updated when you enter the **Point Size** and vice versa. If the **Linked** option is not checked, changing the Point or Pixel sizes will change the font **resolution** in the corresponding fields. You can enter new font horizontal and vertical resolutions. Check the **Proportional** option to keep the horizontal and vertical resolution equal. When the vertical font resolution changes, the Pixel Size also changes while the Point Size of the font remains unchanged. Note that Macintosh screen fonts usually have 72x72 ppi resolution and Windows screen fonts usually have 96x96 ppi resolution.

Leave the **Ascender** and **Descender** fields as they are. You can edit the font metrics later when editing glyphs.

And finally, you need to define the color characteristics for your new font. Choose the font color type in the **Color Mode** dropdown list of the New Font dialog:

Leave the *Black & White* selection if you are creating a common black & white font. Otherwise select the *Grayscale* or the *Color* menu item. In the latter cases you need to choose the number of colors or levels of gray in your font. There are three variants for grayscale fonts: 4, 16 or 256 gray levels; and five choices for color fonts: 4, 16, 256, thousands or millions of colors:



➘ Note: Of course, 16 and 32 bit color tables include grays as separate colors.

After you have described your new font click on the **Create** button. Or click on the **Cancel** button to abort the process. If you press **Create,** a new Font window will open with all the glyph cells gray - this means that no glyphs exist yet in the font. You can create new glyphs by drawing them; pasting them from the Clipboard; or dragging them from other open fonts or images.

There is another way to create a font when you are working within a font project. For example, when you need to create another size or another style from an existing font. Choose the font for transformation in the Project window; drag it to another style by holding down the **Ctrl** key to make a copy or duplicate the font with the **Duplicate** command from the **Edit** menu; and make the appropriate transformations to the new copy of the font.

To change the font size, open the font and use the **Size/Resolution** command from the **Font** menu described in the ***Changing Font Size and/or Resolution*** (on page 149) section of the previous chapter.

To change the font color mode, open the font and use the **Color Mode** command from the **Font** menu described in the ***Changing Font Color Mode*** (on page 151) section of the previous chapter.

Creating a Font from an Image

BitFonter allows you to create a font from bitmap images. The process of converting an image to a font is as follows:

1. Prepare the source image with image editing applications or scan it from within BitFonter.
2. Edit the image with BitFonter.
3. Autosplit the image.
4. Manually correct the autosplitting results.
5. Place the glyphs into a new or open font.

Preparing Images

In BitFonter you can use the following sources of images:

- An image of a font or of individual glyphs printed or written on paper and placed in a scanner
- An image file in TIFF, PNG, BMP or JPEG format
- An image file in vector EPS format
- An image in a format supported by ***ScanFont*** (http://www.fontlab.com/scanfont/)(.sfd, .sf)
- The Windows Clipboard
- The image editor window — where you can draw directly in BitFonter

The easiest way to get an image into BitFonter is to have a scanner connected to your computer.

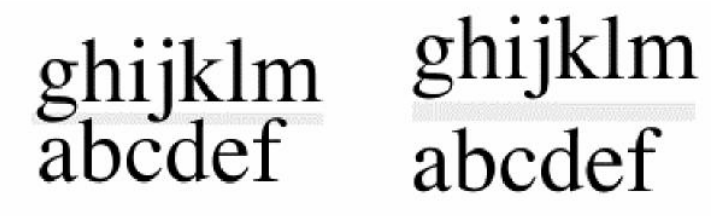
Working in Windows. Place the paper with the image into the scanner and scan it using BitFonter. BitFonter supports the TWAIN interface, so it can work with almost all scanners. (Refer to your scanner's documentation to see whether it will work with the TWAIN interface.) If your scanner cannot work through TWAIN, you can use the program that comes with your scanner to scan the image and save it into one of the formats that BitFonter can read.

If you can place the image onto the Clipboard (using the **Copy** command in any image-editing application), you can get it from there and paste it into BitFonter as a new image.

If you do not have a prepared font image, but are ready to create it using the mouse or a digitizing tablet, you can start from a clear image and draw it using BitFonter's image-editing tools.

In any case, to get the best results, we recommend that you prepare the source image with the following characteristics in mind:

1. If you scan hard copy the paper must be white and smooth.
2. Images of glyphs and symbols must have high contrast.
3. If you are preparing a font of handwriting do not choose a thin pen. A standard 0.7mm pen is best.
4. Try to align the glyphs on a baseline when drawing on paper or in the image.
5. The glyphs' strings must not overlap:



Wrong *Right*

6. Do not overlap glyphs. BitFonter includes advanced algorithms that can separate glyphs, but not when glyphs "touch" each other. Overlapping or touching characters will be treated as just one character. The manual separation of overlapping characters is possible, but is very time-consuming.
7. Consider the resolution of the destination font. The source image resolution must be the same or greater than the font resolution. If the image file resolution is not high enough for your font you will have to increase the size of the characters in the source image or increase the resolution of your scan.

To scan an image from within BitFonter:

1. Place the paper with your image into the scanner;
2. Select the **Select Source** command in the **File > Import** menu;
3. Select the scanner in the dialog that appears;
4. Select the **Acquire** command in the **File > Import** menu;
5. Follow the scanner-dependent scanning dialog to scan the image.

Importing an Image

If you have scanned your image or prepared it with an image editing application, you must save it in a format that BitFonter can open.

BitFonter supports black & white and color images in TIFF, JPEG, BMP or PNG format.

To open an image in BitFonter, select the **Open** command in the **File** menu. In the standard File Open dialog open the folder and select the file that you want to open and click on the **Open** button. You can also drag-and-drop the icon of the image file onto the BitFonter's main window.

You can open several images each in their own window. The number of open images is limited only by the memory allocated to the program.

➲ Tip: When you open an image file the file name will appear in the bottom part of the **File** menu. Later, you can easily open this image again by simply selecting this file name in the menu.

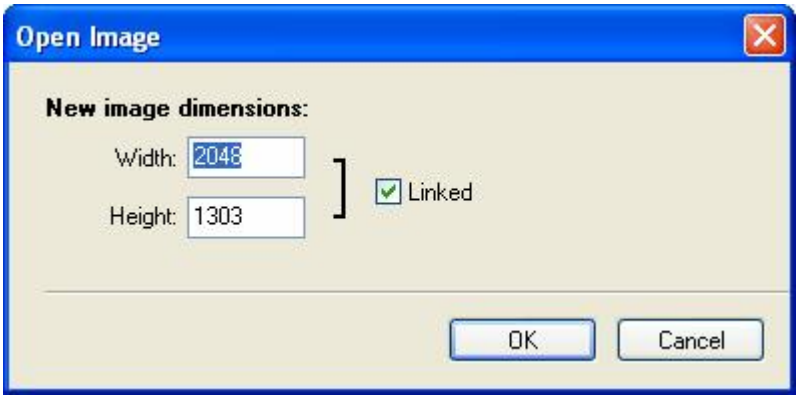
➘ Note: BitFonter has certain limits for the size of the image. The image width cannot exceed 32 Kb. This means true color images (with millions of colors) cannot be wider than 8191 pixels, images with thousands of colors cannot be wider than 16383 pixels, images with 256 colors or grays cannot be wider than 32 K pixels, etc.

Importing EPS

BitFonter can import images in outline ("vector") EPS format. You can save outline images from Adobe Illustrator, Macromedia FreeHand or other vector editors and then use them in BitFonter. Sometimes you may need to change the default options for output in the application's preferences dialog. For example, in Illustrator you should switch on the **AICB** option in the **Options/Files&Clipboard** settings.

To import an EPS file into BitFonter:

1. Select the **Open** command in the **File** menu.
2. In the Open File dialog box that appears, select the .eps file that you want to open and press **Open**.
3. If the imported image contains outlines BitFonter will present the following dialog:



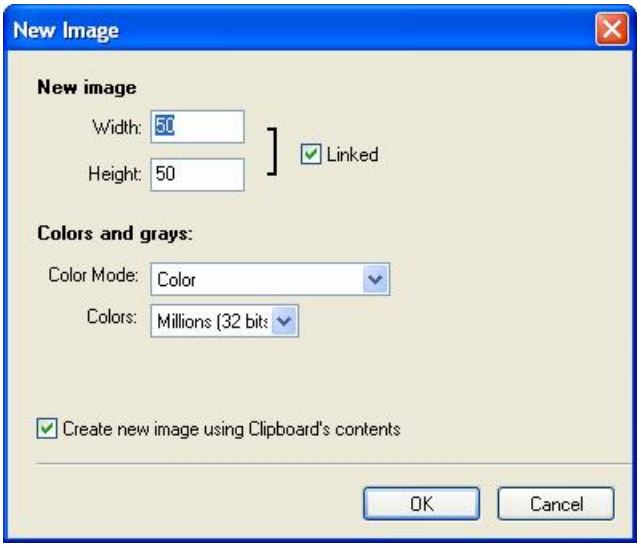
This dialog allows you to control the final image dimensions on import. You may (but do not have to) change the width and/or the height of your image before it opens in the new Image window.

Importing an Image from the Clipboard

Various programs that deal with bitmap images can exchange information by using the Clipboard. If you select and copy an image from such a program, it can be imported into BitFonter. As with image files, BitFonter can import either black & white or color images through the Clipboard.

Importing the Clipboard's contents into BitFonter:

1. Run your image-editing application, open and select the image that you want to import into BitFonter and **Copy** it onto the Clipboard.
2. Switch to BitFonter and select the **Image** command from the **File > New** menu. You will see the following dialog box:



4. Check that the **Create new image using Clipboard's contents** check box is switched on. If this option is not enabled, it means that you do not have a proper image in the Clipboard.

   In the **New Image** section, you will see the size of the image that BitFonter will read from the Clipboard. If you want you can change the image dimensions to resample the imported image.

In the **Color Mode** dropdown list select *Black & White*, *Grayscale* or *Color* for the font you are going to generate from the image. Select the number of colors as well.

5. Click on the **OK** button and you will see a new image window with the Clipboard contents.

Creating an Empty Image

Sometimes you may not have an image ready, only an idea. You can create an image right in BitFonter, using bitmap-editing tools that are of the same quality as in the most well known image-editing programs. But first you have to create an empty image window.

To create an empty image:

1. Select the **Image** command from the **File > New** menu.
2. In the dialog box that appears switch off the **Create new image using Clipboard contents** check box if it is enabled.
3. Enter the dimensions of the new image in pixels in the edit boxes in the **New image** area. Be sure to create the image area big enough to include all the glyphs you want to draw. If you create the image area too small, and later during editing you find that you have no place to include all your drawings, use the **Canvas Size** command in the **Image** menu.
4. Select the color characteristics of the new image as described in the previous section and click on the **OK** button. The *Untitled Image* window will open.

➘ Note: BitFonter has certain limits for the size of an image. The image width cannot exceed 32 Kb. This means true color images (with millions of colors) cannot be wider than 8191 pixels, images with thousands of colors cannot be wider than 16383 pixels, images with 256 colors or grays cannot be wider than 32 K pixels, etc.

Working with an Image

After you scan or import an image from a file or the Clipboard the image will be read and shown in the Image window:

The Image window helps you to preview, edit, and define those glyphs that will be placed into your font. It consists of the image area, Color palette, rulers, scroll bars, the **Zoom** dropdown list, and **Zoom-in** and **Zoom-out** buttons.

The Tools toolbar becomes active when the Image window is open:

Often the size of the image that you open is larger than the size of the window where it appears. You must have a method to view the whole image. You also may need to see the details of an image in a magnified mode. Do this using the **Zoom** dropdown list and **Zoom** buttons located at the bottom of the window. To scroll the image area, use horizontal and vertical scroll bars or the Hand tool by pressing the **Spacebar** key on the keyboard.

To switch on/off the rulers, use the **Rulers** command in the **View** menu.

Some commands available in the Image window can be selected from the context menu. To open the context menu, right-click anywhere in the Image window.

Editing an Image

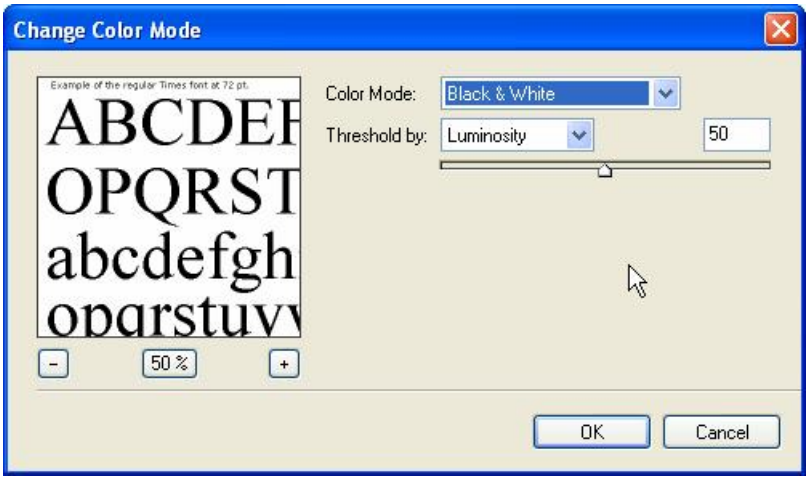
If you want to modify a scanned or imported image, or if you created an empty image and want to draw something, you can use BitFonter's glyph-editing tools and operations. While the Image window is active the selecting and drawing tools are available in the Tools toolbar:

As the techniques of editing the image in the Image window are almost the same as editing the glyph in the Glyph window, we will not discuss them here in this chapter. You will find the full detailed description of all available image-editing tools and operations in the ***Editing Glyphs*** (on page 225) chapter.

Changing Image Colors

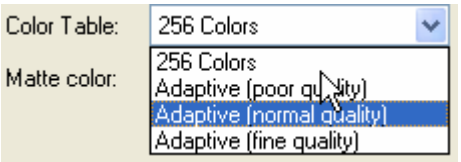
While the Image window is open you can change the image color mode, i.e. the number of available colors. This operation is the same as changing the font color mode described in the ***Changing Font Color Mode*** (on page 151) section.

Select the **Color Mode** command in the **Image** menu and you will see the Change Color Mode dialog:

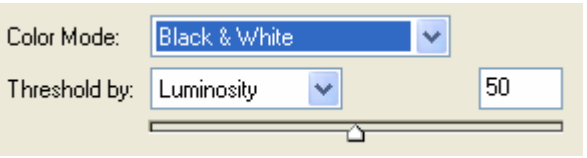


Select a new **Color Mode**, Number of **Colors** and **Color Table** for the image you are editing in the corresponding dropdown lists.

You may want BitFonter to generate a custom color table for your image. This is possible in 4, 16 and 256 color modes only when you change the number of colors in the image. Select one of the available Adaptive color tables:



If you are going to make a black & white image from a color or grayscale image, you must define the threshold parameter for conversion:



See the description of this parameter in the ***Threshold*** (on page 282) section in the ***Editing Glyphs*** (on page 225) chapter.

You can view the preview of the image in the preview field of the dialog box. To see the original image sample, simply press and hold down the **Ctrl** key.

Click on the **OK** button to finish the operation, or **Cancel** to leave the image color mode unchanged.

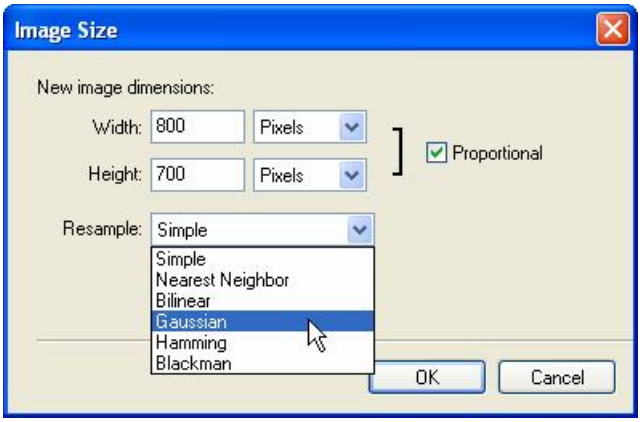
You can undo this operation using the **Edit > Undo** command.

Changing Image Size

Sometimes you may need to change the size of your image. You can do this in two ways.

The first way is to scale the image:

1. Select the **Image Size** command in the **Image** menu. The Image Size dialog will appear:



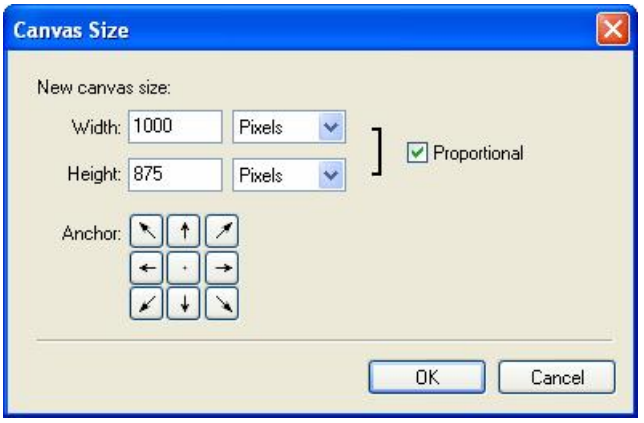
2. Enter the new image **Width** and **Height** in pixels. If the **Proportional** option is checked, the **Height** will be updated while you enter the **Width** and vice versa.

3. There are several predefined methods for scaling images included in the **Resample** dropdown list. If any choice other than the *Simple* resampling method is selected, you can tune it up with the slider to get the best result for your particular image. Note that what is good for one image will not necessarily be good for another.

4. Click on the **OK** button to scale the image, or click on **Cancel** to abort.

The second way to resize an image is adding or removing work space around the image:

1. Select the **Canvas Size** command in the **Image** menu. The Canvas Size dialog will appear:



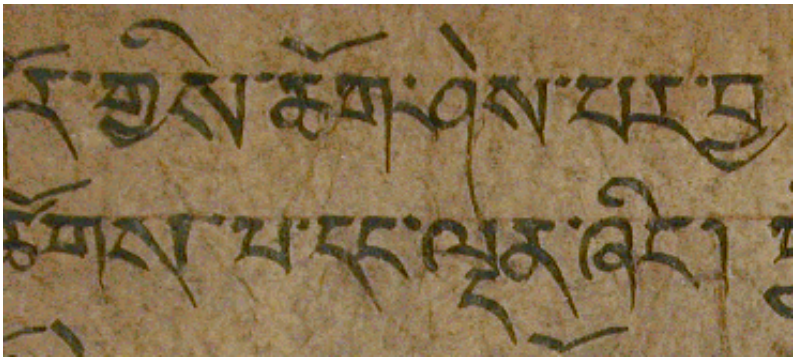
2. Enter the new image **Width** and **Height** in pixels. If the **Proportional** option is checked, the **Height** will be updated while you enter the **Width** and vice versa.
3. For **Anchor**, click on a square to indicate where to position the existing image on the new canvas.
4. Click on the **OK** button to scale the image, or **Cancel** to abort.

You can undo both operations using the **Edit > Undo** command.

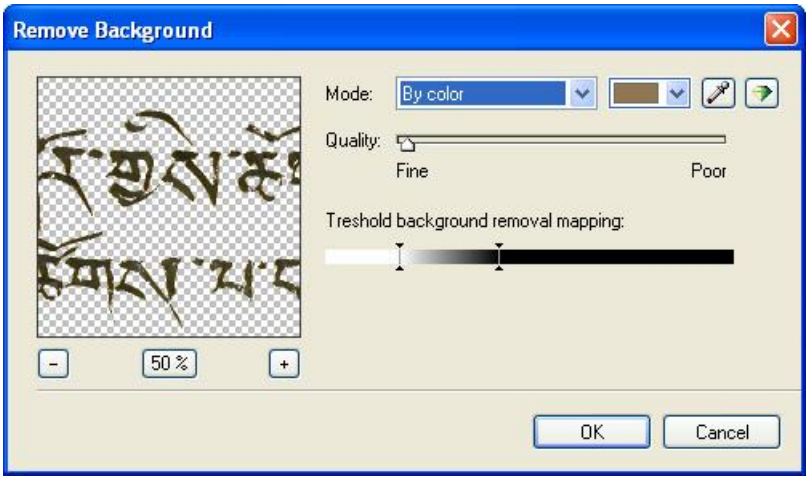
➘ Note: If the image was separated into cells the separation information will be lost when resizing the image.

Removing Background

Sometimes it may be useful to clean up the color background of an image with glyph shapes. For example, you could scan a very old manuscript with a yellow or even brown background:



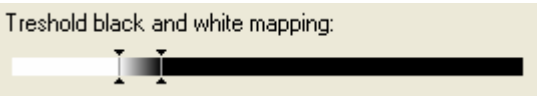
In order to get a font from this image, all the glyph shapes have to be separated from their background. To make this process easier, try using the **Remove Background** command from the **Tools > Filter** menu:



Use the **Mode** dropdown list to select the base algorithm of the operation. You can remove background *By luminosity* or *By color*. You can try them all looking at the preview at the left to choose the best one. The more contrast in the preview the better splitting result you will get.

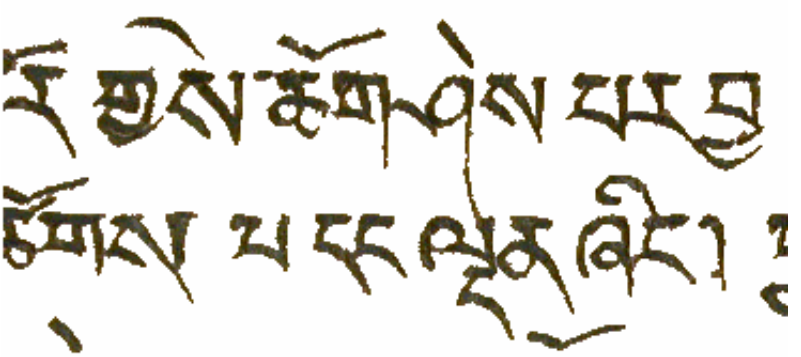
There is also a special tool in color mode. Click on this button and then click on the background in the preview box to pick up a background color.

Use the threshold control for better results:



You can undo this operation using the **Edit > Undo** command.

As a result you will get something like this:



Obviously this image is more useful for splitting into separate glyphs than the original one. But you can continue to enhance the image by using other commands from the **Tools** menu.

Splitting an Image

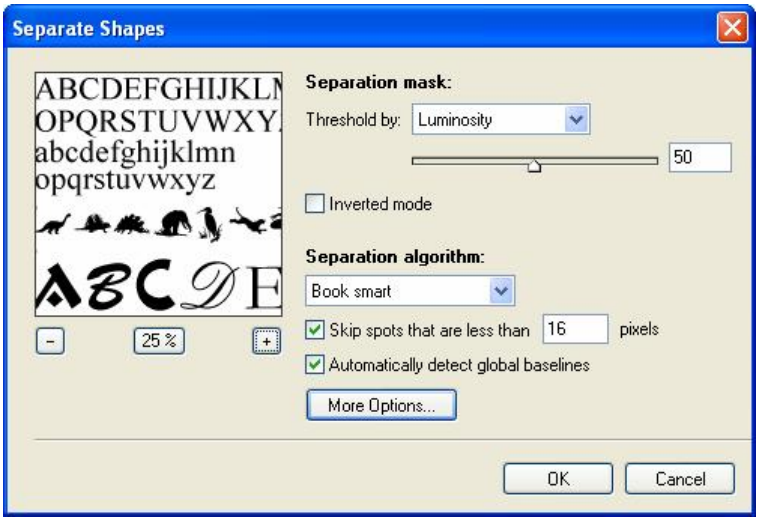
After you edit an image and before you create a font you must split the image into separate shapes. When characters appear on paper (or in a computer-stored bitmap image) they are not “characters” anymore. They are nothing more than a collection of black (or color) spots of various shapes. To restore fonts from an image BitFonter must separate the image of each character from the images of other characters and put this character in the proper position in the collection of glyphs that is called a font.

Because there are differences between the shapes of the same character in different fonts, BitFonter cannot automatically recognize characters, like Optical Character Recognition (OCR) programs do. OCR programs are oriented to recognize a limited number of fonts and use advanced lexical-based algorithms to minimize recognition errors by trying to understand words, not just separate characters. In BitFonter we usually do not have words, just a collection of characters in any order. It is impossible for us to create a recognition algorithm that will understand characters in BitFonter. But we have created tools that simplify manual splitting of glyph shapes and have included a special feature that can automatically locate glyph shapes if they are printed in a known sequence. Our automatic separation feature usually produces very good results requiring few or no manual corrections.

Autosplitting Algorithms

To autosplit an image into glyph shapes:

1. Select the **Separate Shapes** command in the **Image** menu. The Separate Shapes dialog appears:



2. If you are working with a color image, use the **Threshold by** dropdown list and slider to define the boundaries between the white (or whatever color) paper and image pixels. This option works similarly to the ***Threshold*** (on page 282) operation discussed later. You will see the preview of the image changes in the left part of the dialog. Actually the threshold does not change your image or its colors; it is only needed to separate the future glyphs' shapes from their background.

   If you are working with a black & white image, you will see the "short" version of the dialog without the **Separation mask** section and preview.

3. Select the algorithm in the **Separation Algorithm** dropdown list. BitFonter has four autosplitting algorithms to use on different types of images. Here are descriptions with examples to illustrate the differences:

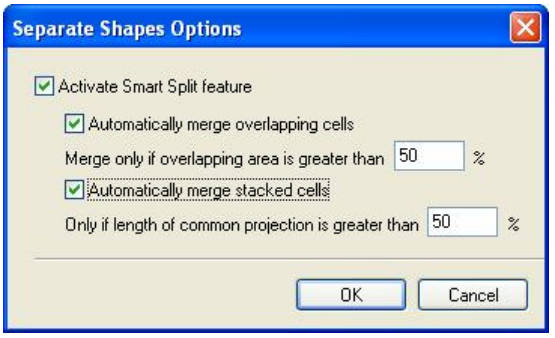
Table with borders		The image has an explicit table structure with cells of equal size separated by vertical and horizontal lines. These lines are treated as the boundaries between the glyphs but will not be added to the glyphs' shapes
Table without borders		The image is the same as above but without vertical and horizontal lines. All the cells are of equal size
Book simple		The cells with separated glyphs are of the same height but have different advance widths. The cells produced by this algorithm cannot overlap each other, so overlapping shapes will occupy one cell which will need to be cut manually later
Book smart		An advanced heuristic algorithm producing cells that can overlap and have different heights and advance widths. It is based on the principle of the "magic wand" (areas selection)

4. Check the **Skip spots that are less than** option and enter the value if needed. This option, when checked, does not treat spots that are below the limit that you enter as glyph shapes or their parts. Such spots are ignored by the algorithm, helping you to avoid speckles of noise that cannot be removed by other operations.
5. Check the **Automatically detect global baselines** option if needed. The autosplit algorithm will locate all the strings and determine the baseline's position in each string automatically. If outlines are present, the baseline's position is calculated according to the outlines.
6. Click on the **OK** button to let BitFonter autosplit the image into cells using the selected algorithm.

If you are not satisfied with the autosplitting results, repeat the steps with another algorithm.

Customizing Autosplit

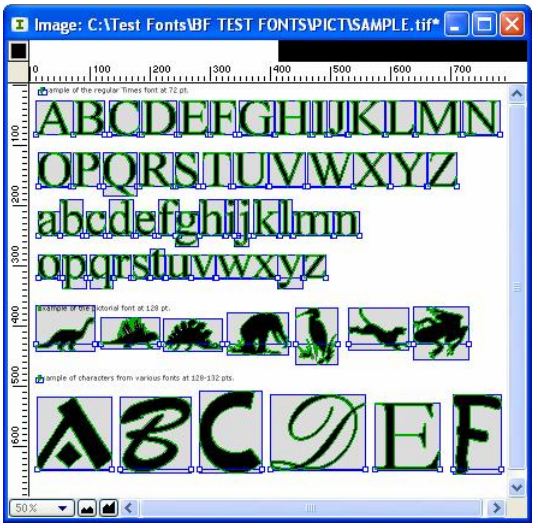
The **Book smart** algorithm is customizable. You can change its parameters by clicking on the **More Options** button in the Separate Shapes dialog. The Separate Shapes Options dialog will open:



The **Automatically merge overlapping cells** option allows you to combine cells that form glyphs like '%' when checked. The value in the editable field defines how much the cells must overlap to be combined into a single cell.

The **Automatically merge stacked cells** option allows you to combine cells that sit on top of other cells when checked. The value in the editable field defines the length of the most common projection in relation to the projection of a smaller cell.

Here is how our sample image looks after autosplitting it with all the default options:



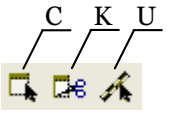
Editing Splitting Data

There are no perfect automatic algorithms. Possible errors of the autosplit algorithm are:

1. If two glyphs touch or overlap each other too much they will be interpreted as a single glyph. These cells must be split manually:

2. Some multi-part glyphs will be interpreted as being separate glyphs. The split cells must be merged.
3. The positions of some of the baselines may be inaccurate. New baselines must be added or unnecessary baselines removed.

The tools for working with the split image are placed at the end of the Tools toolbar and have the following keyboard shortcuts:



Cell Tool

The Cell tool is the main tool for working with the split image. It is used to select and deselect cells for further operations on them. Every cell in the split image will become a glyph in the font that is generated from that image.

To select a cell, click on the Cell tool in the Tools toolbar and then click within the blue rectangle of the cell in the image. Select more cells by shift-clicking or dragging the mouse cursor while you hold down the left mouse button. Use the **Select All** command in the **Edit** menu to select all the cells.

To deselect the selected cells one by one, click on the mouse cursor within the red rectangle of the selected cell while holding down the **Alt** key. To deselect all the cells, click anywhere outside the cells or use the **Deselect** command in the **Edit** menu.

To join two or more selected cells into one cell, select the **Merge Cells** command in the **Image** menu. The selected cells will be merged into one cell, which will remain selected.

To remove unwanted cells (that are not glyphs' shapes, for example), select these cells with the Cell tool and then select the **Delete** command in the **Edit** menu or simply press the **Del** key.

The Cell tool is also used to manually create cells in the image. You can use it to modify the results of automatic splitting or to do manual splitting from the outset.

To draw a new cell:

1. Select the Cell tool in the Tools toolbar and press **Ctrl**.
2. Position the mouse cursor + at the corner of a rectangle to be defined as a cell.
3. Hold down the left mouse button and drag the mouse to draw a rectangle around the glyph in the image.
4. Release the mouse button to finish defining a new cell. A new cell rectangle will appear.

You can define all the glyphs' cells that you need in a font with this manual image splitting tool.

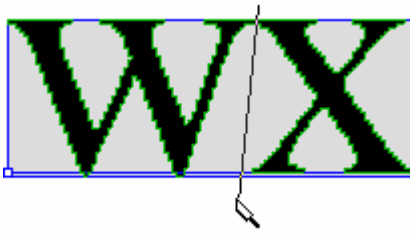
You can define cells with the **Selection** tools as well. Select part of an image with one of the image selection tools and then choose the **Create Cell From Selection** command in the **Image** menu. You will see the selection rectangle become a new cell.

Knife Tool

The Knife tool is used to manually cut a cell into two cells. It is needed when one cell contains two or more glyph shapes after autosplitting the image.

To cut a cell into two cells:

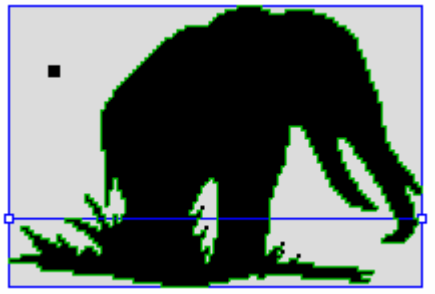
1. Select the Knife tool in the Tools toolbar.
2. Position the mouse cursor at the beginning of the line that will separate the cell.
3. Click the left mouse button to start defining a straight-line segment or hold down the left mouse button to begin a freehand line.
4. Move the mouse to define a straight-line segment or to draw a freehand separating line:



5. Click the left mouse button again to finish defining the straight-line segment or release the mouse button to finish drawing the freehand line. The separating line must intersect the contour of the glyph shape.
6. Double-click the left mouse button to separate the glyph shapes or press the **Esc** key to abort the operation.

Cell Mask

You can see that the glyph shapes in cells have green outlines after autosplitting the image. This is a cell mask. The mask view can be switched on and off with the **View > Show Layers > Contour Mask** command.

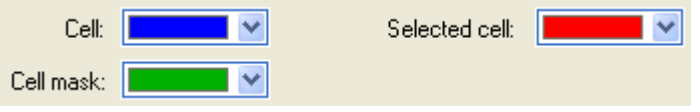


The only purpose of the cell mask is to define which pixels of the split image will go to the glyph in the font. For example, the dot in the top left corner of the picture above is not masked to be included and will be ignored when the cell is exported to the glyph.

Use the **Image > Cell Mask > Update with Selection** command to turn the selected area of an image into a cell mask.

Use the **Image > Cell Mask > Remove** command to completely dispose of the mask in the selected cell. To restore the mask you will need to select image pixels with e.g. the Magic Wand tool and then use **Image > Cell Mask > Update with Selection**.

You can change colors for cell borders and cell mask on the **Glyph and Image Window** page of the Tools > Options dialog described in the ***BitFonter Options*** (on page 44) section:



Saving and Restoring Splitting Data

To save your image with all its separation information use the **Save** command from the **File** menu.

But you may want to save the image splitting data separately.

To save the splitting data:

1. Select the **Save Separation Info** command in the **Image** menu.
2. In the standard File Save dialog box that appears, select the folder where you want to save the data and enter the file name.
3. Click on **Save** to save the data.

Later you can load this saved data and apply it to the currently open image.

To load and apply the splitting data:

1. Select the **Load Separation Info** command in the **Image** menu.
2. In the standard File Open dialog box that appears, select the file with data.
3. Click on **Open** to load and apply the data to the image.

Defining the Scale Factor

The scale factor in BitFonter defines how the glyphs in the split image will fit the future font height. Usually BitFonter automatically computes this scale factor when you separate an image, but it may need to be adjusted especially when cells in the image have different heights. BitFonter has the ability to adjust the scale factor for each of the cells individually or for the whole image.

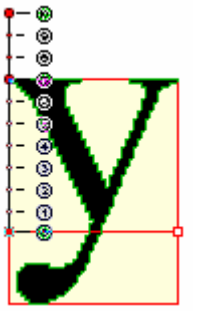
To declare or edit the scale factor for the cells of an image:

1. Click on the **Set Scale** button on the Tools toolbar.
2. The scale bar will appear on the image:

   The scale bar is equal to 100% of the font height. A special highlighted mark on the scale bar is 70%, because usually the height of the uppercase glyphs equals 70% of the font height.

3. Select the cell or a group of cells for which you will set the scale factor and click on the **Auto** button in the Tool Options panel.

If the cell has a baseline, the scale bar's zero point is aligned to the cell's baseline position:

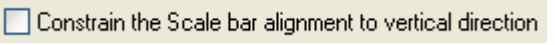


If you are not satisfied with the results of the scale factor's automatic detection, it can be adjusted manually. To do this, position the scale bar so that its length equals the font height.

To move the scale bar, position the mouse cursor on it (not too close to scale bar's ends). The mouse cursor will change to: . Hold down the left mouse button and drag the scale bar to its new place. Release the left mouse button to finish.

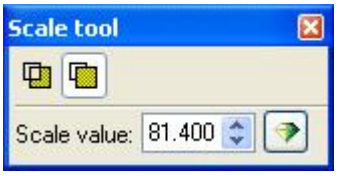
To position the ends of the scale bar, position the mouse cursor on the end that you want to move. The mouse cursor will change to . Hold down the left mouse button and drag the scale bar's end. Release the left mouse button to finish.

To keep the scale bar vertically or horizontally, hold down the **Shift** key on the keyboard. If you want to permanently prevent the scale bar from being slanted, enable this setting in the Tools > Options dialog:



Note that only the length of the scale bar makes sense.

You can see and edit the current scale value (actually the length of the scale bar) in the Tool Options panel. It is measured in image pixels:



The **Auto** button in the Tool Options panel sets the default scale factor for selected cells.

➘ **Note:** If you want to reset all the scale changes you have made for individual cells, deselect all cells and click on the **Auto** button in the Tool Options panel.

Editing Baselines

BitFonter's main purpose is to preserve information. This process consists of several steps:

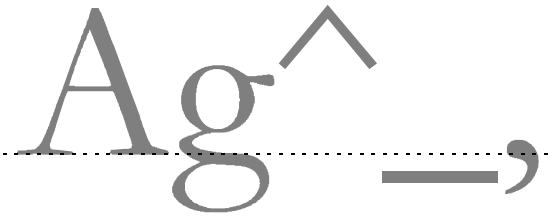
1. Transforming an image into an electronic form (scanning).
2. Determining what parts of the image relate to each glyph (splitting).
3. Determining the actual size of the glyphs (set scale).
4. Assigning the right names and codes to the glyphs (placing them into a font).

At this point you have all the important information restored and are ready to produce a font. However, there are some additional steps that can save time and help you produce better fonts:

A. Analyzing the vertical positions of the glyphs in an image to restore information about the glyphs' vertical position that is necessary to align the glyphs in strings.

B. Analyzing the glyph's horizontal positions to restore metric information.

Step A : Usually, glyphs are aligned in strings in the source image. BitFonter uses this data to restore information about a glyph's vertical positioning. In the string, glyphs are aligned to a baseline — the *zero* level of the string. Some glyphs lie above the baseline, some of them intersect the baseline, and some are below:



The definition of a baseline is the zero point of the vertical coordinates.

In BitFonter you can set baselines for every line of text or even for every glyph cell. When the glyphs are extracted from the image and placed into a font, the information about the relative position of the glyph's image and baseline will be used, so you will not have to reposition the baseline in every glyph. Using baselines in the Image Window can save you a lot of time, as many glyphs are already aligned there. That is why we recommend that you align glyphs in strings when you prepare an original image.

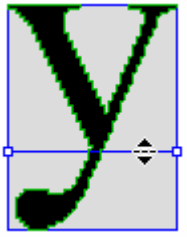
BitFonter automatically detects the positions of all baselines when using the **Separate Shapes** operation. It analyses glyphs' cells and tries to find the best position for the baselines.

If you want to change the positions of the baselines first make them visible.

To make baselines visible in the Image Window, select the **Baseline** item in the **View** menu.

To move a baseline:

1. Select the Cell tool in the Tools toolbar.
2. Position the mouse cursor on the baseline (the cursor changes to).
3. Hold down the left mouse button and drag the baseline where you want it to go. Release the left mouse button to place the baseline in the new position:



To let BitFonter automatically detect the positions of all baselines again without performing autosplitting:

1. Select the Cell tool.
2. Make sure none of the cells selected.
3. Select the **Search Again** command in the **Image > Baseline** menu.

To set the position of a baseline for a string of cells:

1. Select the Cell tool.
2. Select the string of cells by **Shift**-clicking on or rectangular selection.
3. Select the **Search Again** command in the **Image > Baseline** menu.

There are some additional commands in the **Image > Baselines** menu. Here is their meaning:

Set To Cell's Bottom	This command sets the selected cell's baseline (or all baselines if no cell is selected) to the cell's bottom border. This is the same as clearing the cell's baseline
Align To Top	This command sets all the baselines of the selected cells to the position of the highest baseline
Align To Middle	This command sets all the baselines of the selected cells to the position of the calculated average value
Align To Bottom	This command sets all the baselines of the selected cells to the position of the lowest baseline
Auto Align	This command sets all the baselines of the selected cells to the automatically calculated position.

Other Cell Commands

To create a new cell based on the selected area of your image:

1. Using one of the **Selection** tools, select the image portion.
2. Choose the **Make Cell from Selection** command in the **Image** menu.

To select the glyph's shape of a cell:

1. Using the Cell tool, select the cell.
2. Choose the **Make Selection from Cell** command in the **Image** menu.

To set the cell size to the size of the glyph's shape:

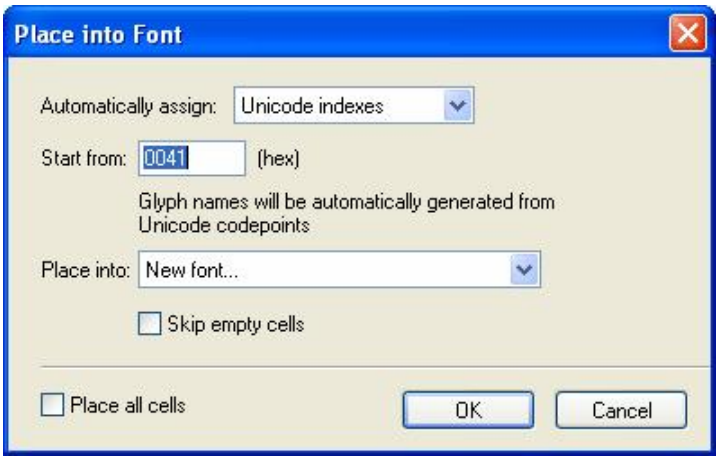
1. Using the Cell tool, select the cell.
2. Select the **Detect Bounding Boxes** command from the **Image** menu.

Creating a Font

The last operation in creating a font from an image is to place the split glyphs into the font. Select those cells that you want to be placed into the font and choose the **Place into Font** command in the **Image** menu.

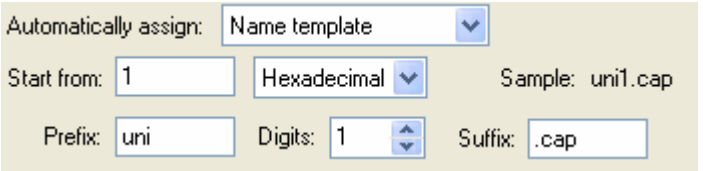
You need not select the cells for placing if you are going to place *all* the cells defined in the image. You may simply double-click on any cell in this case.

The Place into Font dialog will appear:

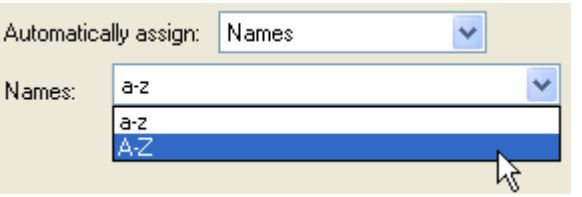


By choosing the *Unicode codepoints* option in the **Automatically assign** dropdown list you make the program assign Unicode indexes to new glyphs. It is also useful to enter the start value in the **Start from** field. Note that you must enter the start value in hexadecimal format. BitFonter will automatically assign names to glyphs by the assigned Unicode codepoints.

By choosing the *Name template* option in the **Automatically assign** dropdown list you can control the naming of exported glyphs. Sometimes it may be useful to enter the prefix and suffix values in the corresponding fields. Note that you can see the sample name at the right:



By choosing the *Names* option you can assign names using the ranges from the dropdown list:



Choose *Nothing* in the **Automatically assign** dropdown list if you do not want BitFonter automatically assign names and/or Unicode codepoints to glyphs.

Leave the *New Font* item in the **Place Into** dropdown list to create a new font or select the font from those available in the menu. Only open fonts will appear in the **Place Into** dropdown list.

Check the **Place all cells** option to place all the defined cells of the image. If no cells were selected this option is not available and all the defined cells will be placed into the font.

➘ Note: Define the order of placement of the image cells and some other options in the Tools > Options dialog described in the ***BitFonter Options*** (on page 44) section.

When you are finished defining cell placement click on the **OK** button. If *New Font* has been selected as the destination, the New Font dialog will be displayed, like when you select the **New Font** command in the **File** menu. Define the main font characteristics as we described in the ***Creating a New Font*** (on page 162) section and click on **Create**. All the cells that you defined for placement into your new font will take their places in the new Font window.

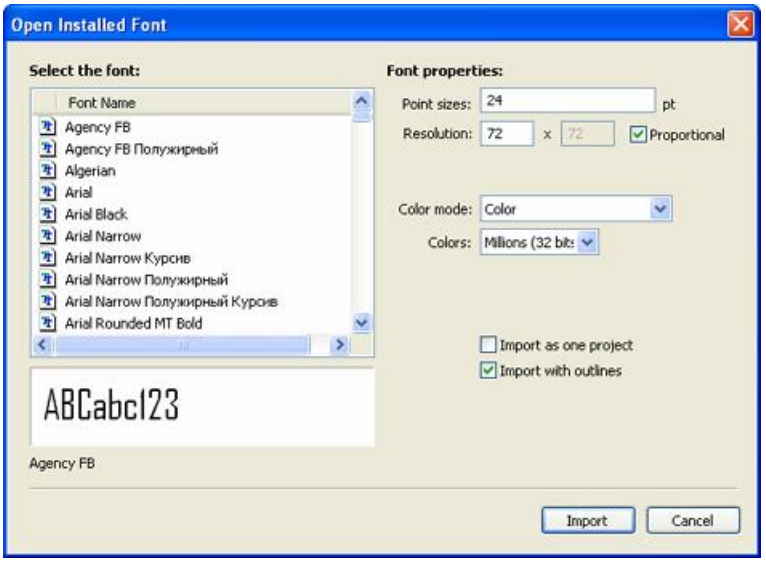
➲ Tip: There is another way to get glyph shapes placed into a font. You can drag-and-drop the cells from the image window right into the Font window.

Importing a Font

It is possible to import fonts from many different sources into BitFonter readable format. **Importing a font** means rasterizing an outline font or getting existing bitmaps and always creates a new font. If you have ***FontLab Studio*** (http://www.fontlab.com/studio/) or ***TypeTool*** (http://www.fontlab.com/typetool/) you can easily import (create) bitmap fonts right from the outline font opened by these applications. To quickly create a font based on one of the fonts installed in your system, use the open installed font feature.

Importing from an Installed Font

To import an installed font, select the **File > Open Installed** command. The Open Installed Font dialog will appear:



The Source Font list contains only outline fonts installed in Windows. Choose any in the list and you will see the font preview in the **Preview** area of the dialog.

BitFonter can generate several bitmap fonts of different sizes at once. Enter the desired sizes in the **Point Sizes** field and the resolution in pixels as in creating a new font. Select the **Color mode**, number of **colors** and the **Color table** if needed as you do when creating a new font.

If the **Import as one project** option is unchecked, new fonts will open in separate Font windows. To combine them in a new project, check this option.

To make BitFonter import glyph outlines, check the **Import with outlines** option. The outline layer can be previewed while working with a glyph's bitmap. Normally glyph outlines will not be imported if the **Import as one project** option was checked. But you can change this: select the *BitFonter Binary Font* format as the default storage format on the **Export** page of the Tools > Options dialog box.

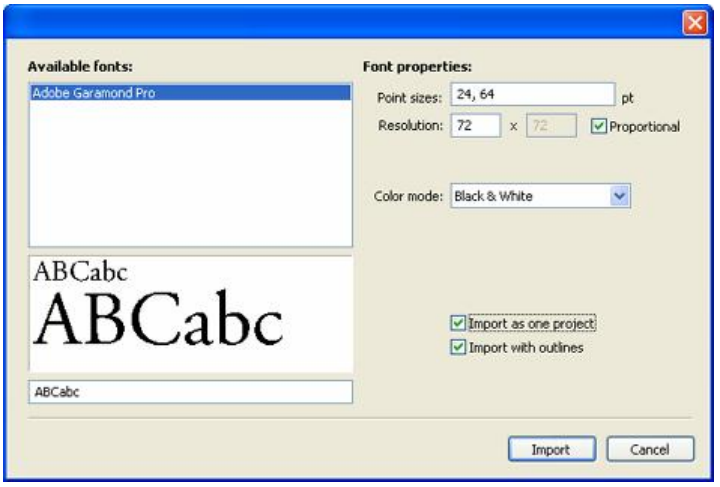
Click on the **Import** button to start import, or **Cancel** to abort the operation. If the **Import as one project** option was checked, you will be prompted with the new project creation dialog.

During the import operation all glyphs of the source outline font are rasterized and placed into the new bitmap fonts. The newly created fonts open in their own Font windows or appear in the new Project window. In the first case glyph outlines, if any, are kept in memory. See the ***Outline Layer*** (on page 233) and the ***Outline Operations*** (on page 300) sections to learn more about outlines in BitFonter.

Importing from Outlines

To import from an outline font opened in ***FontLab Studio*** (http://www.fontlab.com/studio/) or ***TypeTool*** (http://www.fontlab.com/typetool/):

1. Select the **Outline Font** command in the **File > Import** menu and you will see the following dialog:



2. If the **Available Fonts** list is empty, there are no fonts open. Click on **Cancel**, switch to the outline font editing application, open the font that you want to import and return to BitFonter.

3. The **Available Fonts** list now contains all the open fonts. Select one to import. Enter the desired size(s) in points and the resolution in pixels as when creating a new font. Select the **Color mode**, number of **colors** and the **Color table** if needed as you do when creating a new font.

4. The **Import as one project** option allows you to combine several fonts of different sizes in a new project. If it is checked, you will be prompted to create the project with the standard Save dialog.
5. To make BitFonter import glyph outlines, check the **Import with outlines** option. The outline layer can be previewed while working with a glyph's bitmap. **Outlines will not be imported** if the **Import as one project** option was checked and the *BitFonter Binary Font* format was not selected as the default storage format.
6. Click on the **Import** button to import the font.

The outline glyphs will be rasterized and imported from the server application to BitFonter's new fonts - one for each point size. Every font will be opened in its new Font window. If the **Import as one project** option was checked, all the fonts will be added to the new project and saved to the disk.

Saving a Font

While working with an open font you can save your work to disk anytime you want.

To save the font in the same file and same format as it was opened, select the **Save** command in the **File** menu or click on the button in the Standard toolbar. Save the font project also if the font that you have just saved is a part of the project. If you have saved a font as part of a project by mistake, revert the project to the last saved using the **Revert** command in the **File** menu.

➘ Note: You will be asked to save any unsaved project fonts when saving a project. In other words, all the fonts that are part of a project must be saved before the project will be saved.

To save your font under a different name or in another format, use the **Save As** command in the **File** menu. When you select this command the standard Save As dialog appears. You can select the font format in the **Format** dropdown list located at the bottom part of the dialog. Enter the file name and select the destination where you want to save your font. After saving the font with the **Save As** command, this new font remains open in its window.

The range of available formats depends on the font's color mode. See the ***Saving Color Fonts*** (on page 210) section for details.

All the possible formats are:

BitFonter Binary Font (.bfb)	A special format for BitFonter 3. It has the same function as the .vfb format in ***FontLab Studio*** (http://www.fontlab.com/studio/) or ***TypeTool*** (http://www.fontlab.com/typetool/).
BDF Font (.bdf)	The Adobe™ Glyph Bitmap Distribution Format (BDF), which is intended to be easily understood by both humans and computers. BDF files are ASCII encoded and may be read as text. BDF is the most universal bitmap format
HP Soft Font (.sfp)	Hewlett Packard Soft Font format for use in the LaserJet family of printers
Microsoft Windows Font (.fnt)	MS Windows screen font
AFP Font Resource (.afp)	The format for font resource data to be loaded and managed by Advanced Function Printing (AFP) software. This is one of the public font information interchange formats
Palm OS font (.pdb)	A special font format for use in Palm pocket computers. Fonts in this format are always opened as a project
PhotoFont® (.phf)	A special font format where every glyph is represented by an image. Unlimited font color mode makes this format useful for clipart purposes in web and similar design. A font in this format is always opened as a project
FontLab Template (.dat)	The template font file used when working in FontLab

➘ Note: Glyph outlines that were imported or created manually can be saved only in BitFonter Binary Font (.bfb) format.

Saving in BitFonter Binary Font Format

BitFonter Binary Font (.bfb) format was developed especially for BitFonter. This bitmap format is the most universal since it may contain any number of glyphs in any color mode of any size/resolution with any names, codes, and Unicode codepoints or without them. The Outline layer can also be stored in files of this format. You can use the BitFonter Binary Font format to save your intermediate results while working with any bitmap font regardless of the final font format.

BFB format was chosen as the main BitFonter format. When saving the project in a .fnp file, BitFonter saves all the fonts included in the project in the BFB format (this can be changed on the **Export** page of the Tools > Options dialog).

The .bfb format in BitFonter has the same function as the .vfb format in ***FontLab Studio*** (http://www.fontlab.com/studio/) or ***TypeTool*** (http://www.fontlab.com/typetool/).

Saving in BDF Format

Bitmap Distribution Font (BDF) format may contain any number of glyphs in any color mode of any size/resolution with any names, codes, and Unicode codepoints or without them. The fonts in this format may have custom color tables. You may use the BDF format to save your intermediate results while working with your new fonts or font sets.

➘ Note: The Outline layer cannot be stored in the BDF format. Use the BitFonter Binary Font format if you want to preserve outlines.

When saving in BDF format, the default export options are used.

These options are:

Generate XFREE86_GLYPH_RANGES fields	When this option is on, BitFonter will generate BDF files with special key words that define existing glyphs and their codes. Usually this option need not be switched on
Pad Cells	When this option is on, BitFonter will generate and save glyphs of equal size in BDF format. Usually this option need not be switched on
Write encoding codepoints instead of Unicode codepoints	When this option is on BitFonter writes 1-byte codes for glyphs but not 2-byte Unicode codepoints. This option is off by default
Line terminator	This dropdown list lets you define the end of line style in a BDF file. You may choose among Mac, DOS and Unix line terminators. DOS EOL mark is the default value.

Use the **Export > BDF/ABF** page of the Tools > Options dialog to change BDF export options if needed.

Saving in HP Soft Font Format

BitFonter can save fonts in Hewlett Packard Soft Font format (.sfp) for use in the LaserJet family of printers. Only black & white fonts can be saved in this format. The font cannot have glyphs with duplicate or negative codes and the glyphs' names and Unicode codepoints are not written in this format. The font cannot have more than 256 glyphs and cannot have glyphs larger than 256x256 pixels.

When saving in HP Soft Font format, the default export options are used.

These options are:

Format	Choose the type of printer for which your font is generated
Font type	Choose among 7-bit, 8-bit and PC-8 options
Orientation	Select Portrait or Landscape orientation (whichever the font is designed for)

Use the **Export > HP Soft Font** page of the Tools > Options dialog to change HP Soft Font export options if needed.

Saving in Windows Font Format

BitFonter can save fonts in Windows Font format (.fnt) for use on Windows 3.1 and later. Only black & white fonts may be saved in this format. The font cannot have glyphs with duplicate codes, negative codes or without codes. The glyphs' names and Unicode codepoints can be written in this format but are not necessary. The font cannot have more than 256 glyphs and cannot have glyphs larger than 256x256 pixels.

Saving in AFP Font Resource

BitFonter can save fonts in Advanced Function Printing (AFP) resource format. Only black & white bitmap fonts may be saved in this format. The font can have glyphs' names, codes and Unicode codepoints in this format but may not. Unicode codepoints are not written in this format. The number of glyphs is not limited. BitFonter limits the size of glyphs to 2048 by 2048 pixels.

Saving in Palm OS Font Format

The Palm OS font format (.pdb) has many restrictions as well. Only black & white fonts can be saved in this format. The font cannot have glyphs with duplicate codes and the glyphs' names and Unicode codepoints are not written in this format. The font cannot have more than 256 glyphs and cannot have glyphs larger than 256x256 pixels. Although you may save a project with multiple families in this format, the font will contain only one family. This font format contains less font information; therefore saving in this format is an export with significant information loss.

Note that this format is of a project type in BitFonter.

➘ Note: The pixel size of any Palm OS font is always defined by the font ascender value.

Saving in PhotoFont Format

A font in PhotoFont (.phf) format contains glyphs as a set of true color bitmap images. You can save any existing font of any bitmap format in this format. A project containing an outline font or fonts of more than one style cannot be saved as a PhotoFont font.

There is only one restriction on the glyphs in this format: a font of any color mode will be saved as a 32-bit true color font. This is the reason the .phf font file is 10 times bigger than its black & white twin. The PhotoFont format accepts glyphs' names, codes, and Unicode codepoints but not all of them are necessary. Fonts in the PhotoFont format can be kerned. All these features make this format useful for incorporating images in font design.

Note that this format is of a project type in BitFonter.

Saving as a FontLab Template

The FontLab Studio 5 Template font (.dat) is used in Fontlab applications (including BitFonter itself) for showing a font chart template in the Font window. This special template font usually contains several thousand characters that cover almost all Unicode codepoints. A font in this format may contain only black & white glyphs with their Unicode codepoints assigned. The indexes cannot have duplicates. The size of the glyph's bounding box must be 24x24 pixels. The glyphs' names and codes as well as other possible font information are not needed and not written in this special format.

The *template.dat* file that is used in Fontlab applications is installed in the [Shared default data]\Data\ folder.

Saving Color Fonts

When saving a font in a specified format, you should know the restrictions mentioned in the previous section. This section summarizes those restrictions concerning the color fonts:

1. Color fonts with 4, 16, or 256 colors may be saved as a BDF Font file or PhotoFont only.
2. Grayscale fonts with 4, 16, or 256 levels of gray may be saved as a BDF Font file or PhotoFont only. In addition, bitmap fonts with 4 or 16 levels of gray may be saved as an embedded part of a Microsoft Windows OpenType SBIT Font.
3. Color fonts with thousands and millions of colors may be saved as BDF or PhotoFont files only.

Use the table below to quickly see the available font formats for each color mode:

Font Color Mode	Bits per pixel	Font file formats
Black & white	1	BDF Font; HP Soft Font; Microsoft Windows Font; OpenType SBIT Font; AFP Font Resource; Palm OS Font; PhotoFont*; FontLab Template
4 colors	2	BDF Font; PhotoFont
4 levels of gray	2	BDF Font; Microsoft Windows OpenType SBIT Font; PhotoFont*
16 colors	4	BDF Font; PhotoFont*
16 levels of gray	4	BDF Font; Microsoft Windows OpenType SBIT Font; PhotoFont*
256 colors	8	BDF Font; PhotoFont*
256 levels of gray	8	BDF Font; PhotoFont*
Thousands of colors	16	BDF Font; PhotoFont*
Millions of colors and a transparency layer	32	BDF Font; PhotoFont*

*Actually, the PhotoFont format is always true color (32 bits) and it will be saved and opened as a 32-bits font regardless of the original font color mode.

As mentioned above, BitFonter Binary Font (.bfb) format is the most universal. Anything that you can create and edit in BitFonter can be saved in this format. That is why we suggest you save your intermediate results in this format. When you are finished editing your font, you can choose the final font format that you need.

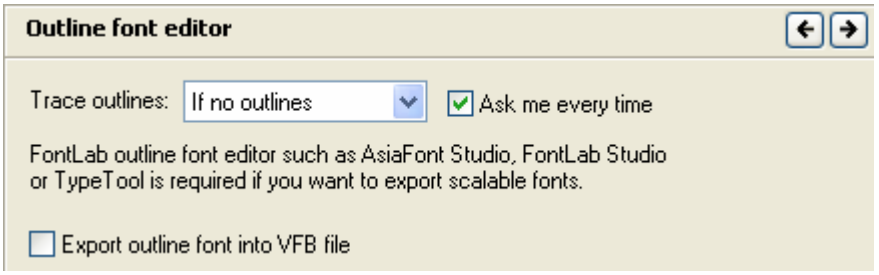
Exporting a Font

There are four types of export features in BitFonter: export to outline font, export to pixel font, export to a single image, and export to a set of images. We do not discuss saving fonts in different font formats here. Saving fonts was discussed in the ***previous section*** (on page 203).

Outline Font Editor

As you know, BitFonter has no outline font-editing features, so you need another program to accept outline glyphs from BitFonter. We call programs that can accept data from BitFonter or other sources “Outline font-editing Applications”. FontLab 4.x, FontLab Studio 5, AsiaFont Studio, and TypeTool 2+ are programs that can establish communication with BitFonter; accept data in outline and bitmap form; and report to BitFonter the results of the data transfer so it can react properly.

Please note that you must have any of the listed applications installed together with BitFonter in order to create outline fonts. If you do, you can customize the communication options on the **Outline font editor** page of the Tools > Options dialog:


Outline font editor
Trace outlines:
If no outlines
Ask me every time
FontLab outline font editor such as AsiaFont Studio, FontLab Studio or TypeTool is required if you want to export scalable fonts.
Export outline font into VFB file

See the detailed description of these options in the ***Outline Font Editor*** (on page 62) section of the ***BitFonter User Interface*** (on page 13) chapter.

If you do not have any of the listed applications installed you still can choose to export outlines into VFB file — check the **Export outline font into VFB file** option.

Exporting Outlines

When exporting a font to an outline font-editing application, BitFonter performs several actions:

- Prepares glyphs for tracing.
- Traces glyphs in accordance with the tracing parameters.
- Prepares the font information that is needed for outline font generation.
- Finds and launches the outline font-editing application.
- Transfers the glyphs' outlines and font information to the outline font-editing application.

To start the export procedure:

1. Select the font or fonts in the Project window or select glyphs in the open Font window.
2. Select the **File > Export > Outline Font** command. If at least one glyph in your exported font has an outline layer BitFonter will show you the Export as Outline Font dialog asking what should be done with existing outlines:

Export as Outline Font
Trace outlines if they're missing
Always trace outlines
Export existing outlines only
Trace options...
Always apply these settings for current document
If this option is on, hold down the Control key while choosing the menu command to open this dialog
Export
Cancel

There are several radio buttons, a check box, and the **Trace Options** button in the dialog. If some glyphs of the font have no outline layer, you can tell BitFonter to create these outlines by choosing the **Trace outlines if they're missing** option. If you select the **Always trace outlines** option, the existing outlines will be ignored and new outlines will be created and exported. If you want BitFonter to **export existing outlines only**, then choose the third radio button.

If you want BitFonter to use the selected option without asking you, then check the **Always apply these settings for current document** check box. You will see this dialog box again if you hold down the **Ctrl** key when selecting the **File > Export > Outline Font** command.

3. If any glyph in your exported font has no outline layer or when you click on the **Trace Options** button in the Export as Outline Font dialog BitFonter opens the Trace Options dialog described in the ***Trace Options*** (on page 302) section. Set the options for tracing and click on **OK**.

4. When finished, press **Export** to proceed, or **Cancel** to stop the procedure. If the **Export outline font into VFB file** option is off BitFonter will launch outline font editor and present another dialog:

Select the font opened in FontLab Studio (or TypeTool, or AFS) in the **Place into** dropdown list or choose *New font*. Switch on the **Export entire font** option if needed and click on **Export** to finish.

If **Export outline font into VFB file** was switched on in Tools > Options BitFonter will open the standard File Save dialog box. Select the folder where you want to save the .vfb file and enter the file name.

The bitmap glyphs will be converted to black & white mode (if not already), autotraced and transferred to the outline font-editing application (which will be launched if not already open) or exported in the vfb format. If the **Export existing outlines only** option was checked and there were glyphs with outlines in the font, these outlines will be transferred to the outline font-editing application (or exported to vfb) directly and the trace options will be ignored. The exported font will open in a new FontLab Studio (or TypeTool, or AFS) Font window, where you can start working with the outlines.

➘ Note: The options that you define in the Trace Options dialog are used also when you select the **Trace Bitmap** operation described later in the ***Outline Operations*** (on page 300) section.

Exporting to Outline Pixel Font

With BitFonter and ***FontLab Studio*** (http://www.fontlab.com/studio/) you can create new outline pixel fonts in minutes.

What Are Pixel Fonts?

Pixel fonts (also sometimes known as "Flash fonts") are TrueType or Type 1 outline fonts especially designed for low resolution screen display. Regular outline fonts are designed primarily to be printed at high resolution. This is not the purpose of pixel fonts.

A monitor's screen has a grid of small pixels. The glyphs in a pixel typeface are made from "pixels" too. Every "pixel" in each glyph design has to correspond exactly with a monitor's pixel.

When you view a regular outline font on the screen, you see the result of the system rasterizer's work. The rasterizer is the program that converts glyph outline information into pixel information for the monitor. I.e. it tells the monitor which pixels to turn on and off to make the glyph's image on the screen. Regular fonts will look fine at medium to large sizes, 12 points and larger, especially if font smoothing (anti-aliasing) is switched on, but as the font size gets smaller the anti-aliasing blurs the glyph making it fuzzy and hard to read:

Bitmap and outline pixel fonts are designed so that they look good at small sizes without needing font smoothing:

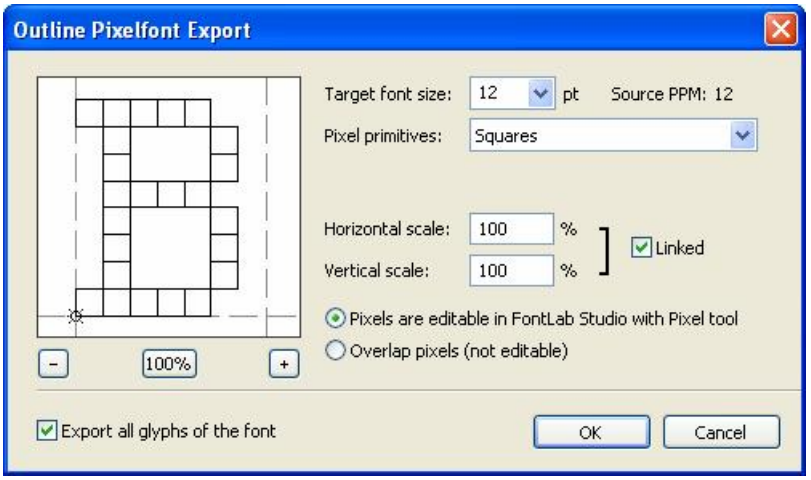
Pixel fonts are widely used for web page design, especially in Macromedia Flash. In all other respects they are typical outline fonts that can be installed on a Windows or Mac OS system without any additional software. Generally pixel fonts can be created in any font format - TrueType/OpenType TT, Type 1, OpenType PS - but the TrueType/OpenType TT font format is more common.

Pixel fonts are always created at some specified size (e.g. 8 pixels). This is merely a setting of the font and not necessarily its actual size. However, it is generally the size at which the font will look best. When using a pixel font you should always keep in mind its size. Multiplications of the specified size will also work (e.g. 16, 24, 32 etc.) but will look pixelated at larger sizes:

The Exporting Process

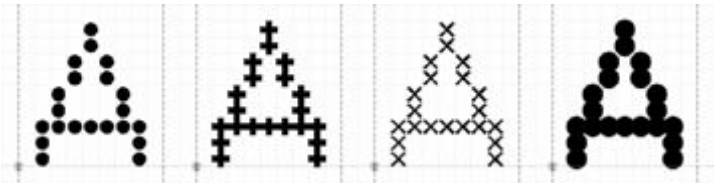
To export all the font glyphs or selected glyphs only:

1. Open the font in the Font window in BitFonter and make sure it is black & white. If not use the **Font > Color Mode** command to change the font colors to black & white.
2. Select the cells with glyphs that you want to be exported.
3. Choose the **Outline Pixelfont** command in the **File > Export** menu. BitFonter will show you the following dialog:



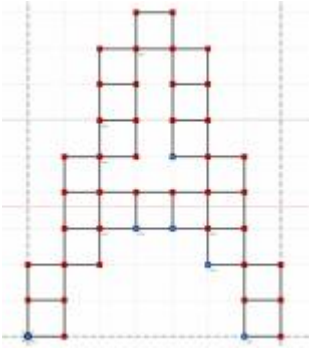
4. The **Source PPM** value tells you the original size of the bitmap font in pixels. Set the **Target font size** for your new pixel font. We recommend that you set the target font point size equal to the source font PPM size for best results.

5. Set the desired shape of "pixels" in the output font using the **Pixel primitives** dropdown list. The normal shape is *Squares* but you may need to change it if you want to get out of the ordinary results. Here are some examples:

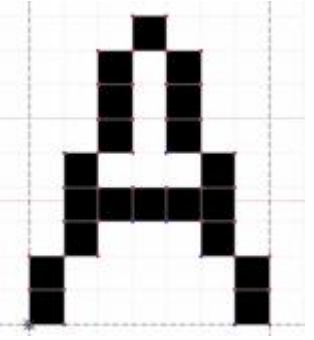


6. Scaling of the pixel elements is also an optional feature. The common values are 100x100%, i.e. without scaling pixels.

7. **Make pixels editable in FontLab Studio** (or **Overlap pixels**). If you select **Pixels are editable** the output glyph image will consist of several separate square (or whatever shape selected) contours each representing a pixel:



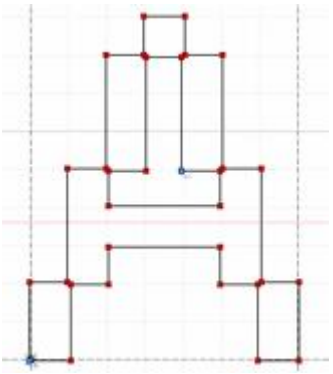
Unfilled contours



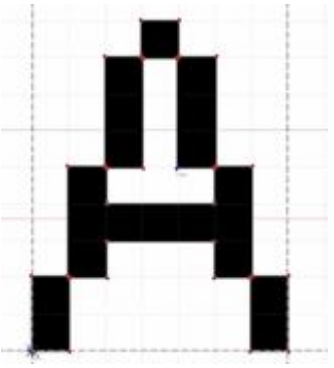
Filled contours

If you have the Python system installed you will be able to edit such glyphs with the special Python tool in ***FontLab Studio*** (http://www.fontlab.com/studio/).

Choose the **Overlap pixels** option if you are not going to edit your font with the Python tool. BitFonter will generate contours that can be edited in FontLab Studio (or ***TypeTool*** (http://www.fontlab.com/typetool/), or ***AFS*** (http://www.fontlab.com/asiafontstudio/)) in the usual way but will not be editable by the Python tool:



Unfilled contours



Filled contours

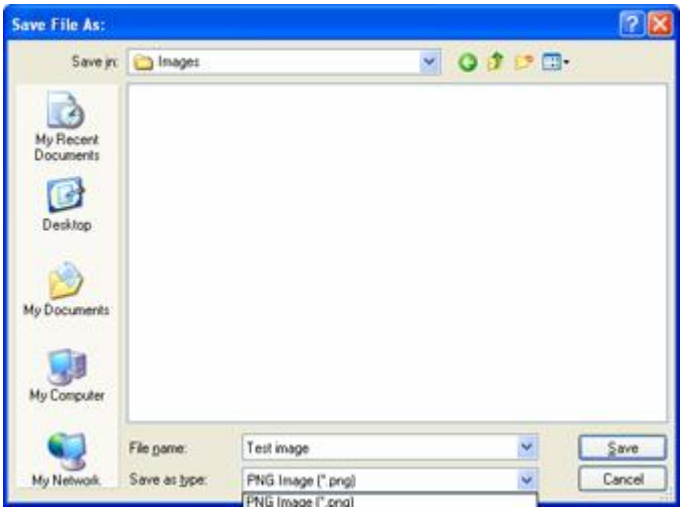
8. If any glyphs were selected, you may switch off the **Export all glyphs of the font** option to export only selected glyphs. Click on **OK** to start exporting, or **Cancel** to stop the procedure.

After BitFonter finishes, the font will open in your outline font-editing application. Note that BitFonter does not export the font in TrueType or Type 1 format, this is done in TypeTool/FontLab Studio/AsiaFont Studio.

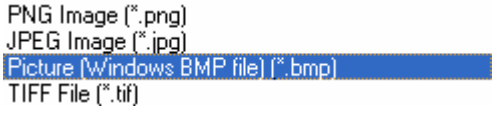
Exporting to Image File

To export all the font glyphs or selected glyphs only to a single image file:

1. Select the font or fonts in the Project window or select the glyphs in the open Font window.
2. Choose the **Image** command in the **File > Export** menu. Name the file, choose image format and select its destination in the standard Save File As dialog box:



You can choose among PNG, JPEG, TIFF or BMP image in the **Save as type** dropdown list:



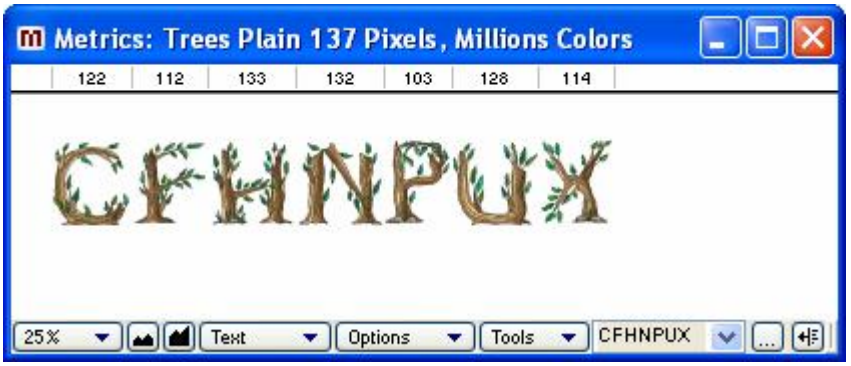
Click on **Save** when ready.

3. If glyphs were not selected, you will be asked whether you want to export the whole font. Press **Yes** to start exporting, or **No** to stop the procedure.

➘ Note: The fonts selected in the Project window will be exported as separate images.

There is another way to get an image of your font:

1. Make sure only glyphs that you are going to export are selected in the Font window.
2. Choose the **Metrics Window** command in the **Window** menu. You will see the Metrics window containing the selected glyphs only:



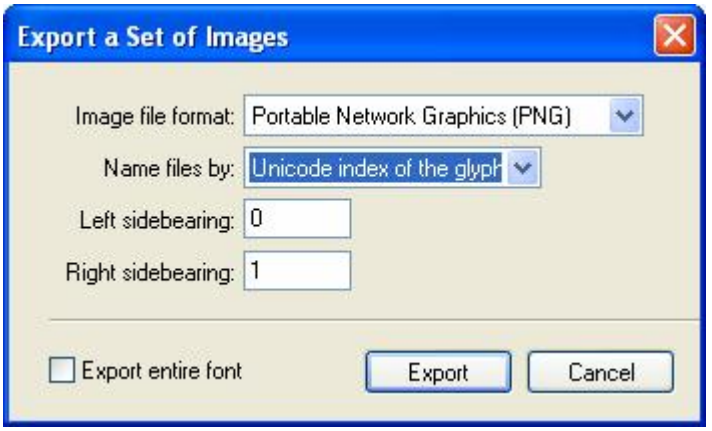
   You can edit the string of glyphs here as you edit text in a text editor. You can delete characters with the **Backspace** key.

3a. Choose the **Image** command in the **File > Export** menu; name the image file; select the file format (BMP, JPEG, TIFF or PNG image) and a destination for it in the standard Save As dialog. The glyphs you have placed in the Metrics window will be saved as a table in an image file.

3b. Alternatively choose the **Export Preview as image** command in the local **Tools** menu at the bottom of the Metrics window. The glyphs you have placed in the Metrics window will be saved as a string in an image file.

Exporting as a Set of Images

You may choose to save your font glyphs as a set of separate image files. To do this, select the font or fonts in the Project window or select the glyphs in the open Font window and then choose the **File > Export > Set of Images** command. This procedure is a little more complicated than the export to a single file. The Export a Set of Images dialog will appear:



You can define the following export parameters before clicking on the **Export** button:

Image file format	Select the image format for export: Portable Network Graphics (PNG) format or BMP format
Name files by	Select the file naming principle here. The image files can get their names from the glyph names or Unicode codepoints
Left sidebearing **Right sidebearing**	Define the white margins to the left and right of the glyph shape in pixels here. Set these fields to zero if you want the width of image to be equal to the advance width of the glyph shape
Export Entire Font	Check this option if you want to export all font glyphs or leave this option unchecked to export the selected glyphs only. If you have selected the font in the Project window, the entire font will be exported anyway

When you are ready, click on the **Export** button. The standard dialog will appear to let you choose the destination for the files. Every glyph will be saved as separate image in one folder.

Editing Glyphs

When you create a brand new font you will draw all the glyphs that you want to be included in the font. Also, when you generate a bitmap font from outlines or create glyphs by image importing, you may want to edit them to get the best result. Drawing and editing an individual glyph is a very hard and time-consuming job, but it is the only way to get the perfect font.

BitFonter includes professional bitmap glyph-editing tools and operations for this purpose. It is time to review them.

Most of the editing tools and operations discussed here are equally applicable to scanned, imported or newly created images. You can work with a whole image as if it is a big glyph.

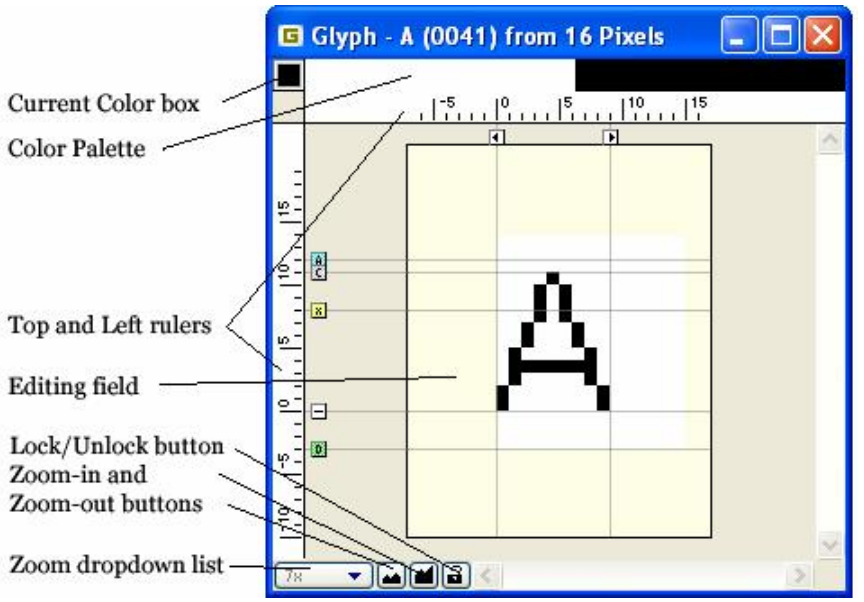
The Glyph Window

The Glyph window is a window for glyph shape drawing and editing. You can easily open the Glyph window by double-clicking on a glyph cell in the Font window. Double-clicking on a non-existent glyph (gray empty cell) creates a glyph and opens an empty Glyph window for drawing.

If the **Open glyph in new window** option on the **Font Window** page of the Tools > Options dialog is checked (see the ***Font Window Options*** (on page 48) section) the glyph will be opened in a separate Glyph window. If the **Open glyph in new window** option is unchecked, the glyph will be opened in the existing Glyph window. Holding down the **Ctrl** key when double-clicking reverts the meaning of this option.

If the **Double-click creates an empty glyph first** option on the **Font Window** page of the Tools > Options dialog is unchecked (see the ***Font Window Options*** (on page 48) section), double-clicking on a non-existent glyph creates a space for a glyph and opens an empty Glyph window for drawing. Otherwise you need two double-clicks: to create the glyph and to open it in the Glyph window.

The Glyph window is a little more complex than the windows that we discussed earlier:



The Glyph window consists of the following parts and controls:

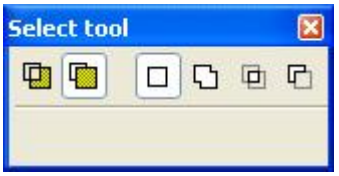
- Editing Field
- Top and left rulers
- Color Palette
- Current Color box
- Scroll Bars
- Zoom dropdown list
- Zoom-in and Zoom-out buttons
- Lock/Unlock button

The Tools toolbar becomes active when the Glyph window is open:

The tools are divided into four groups:

- selection tools
- drawing tools
- tools for working with cells in an Image window (not available when in the Glyph window)
- others, including the Eye Dropper tool, Metrics tool, and Zoom tool.

The **Tool Options** panel also becomes available:



The panel's title depends on the currently selected tool.

You can switch toolbars and panels on and off using the commands in the **View** menu.

Glyph Structure

All glyphs consist of three information groups:

1. Information about the glyph shape (or bitmap layer).
2. Information about the glyph and font metrics.
3. Information about the glyph outline layer.

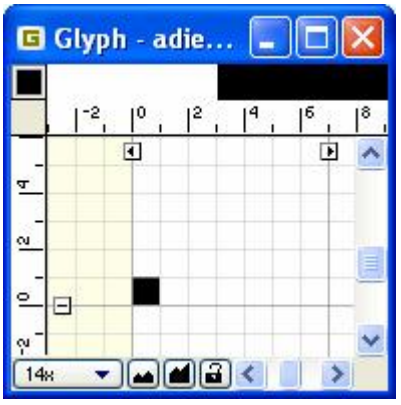
All this information may be changed in the Editing Field.

Units of Measurement

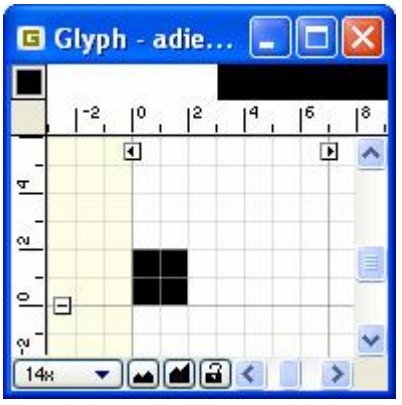
The top and left rulers of the Glyph window are measured in pixels. The pixel is the minimal unit of a bitmap font. All font measurements are done in this unit. In Mac OS bitmap fonts are created as screen fonts at 72 pixels per inch resolution (96 pixels on Windows), and in this case (and only in this case) we may speak about a font pixel as a screen pixel. In other cases (e.g. on Windows) a pixel in the glyph shape and a screen pixel are not the same.

Similarly, the glyph pixel is equal to a glyph point at 72 pixels per inch resolution (on Macintosh). But if, for example, vertical and horizontal font resolution is 144 pixels per inch, every glyph point consists of four glyph pixels.

Point = 1 pixel at 72 ppi



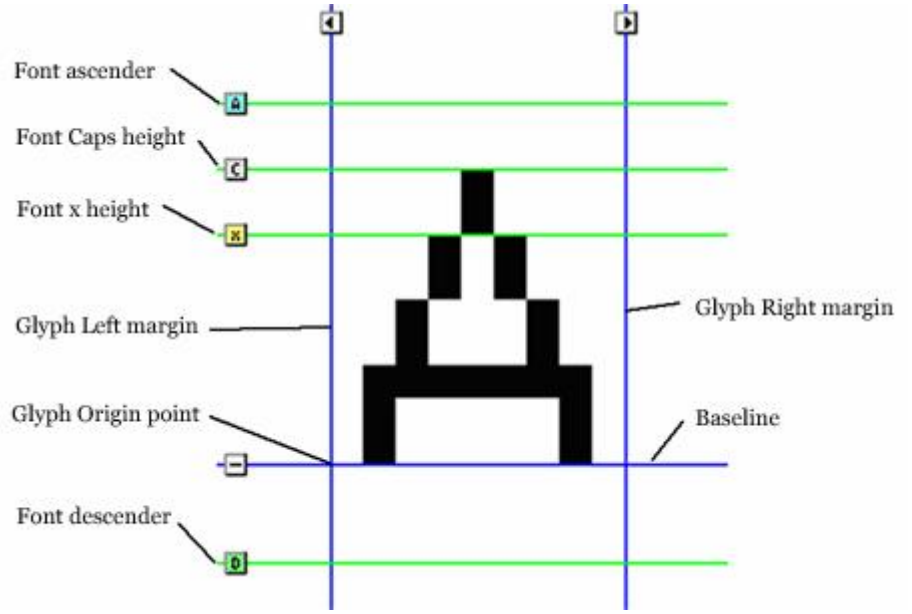
Point = 4 pixels at 144 ppi



➘ Note: In a Project window bitmap fonts are named by their pixel size not the point size.

Metrics

The Metrics data of a glyph includes information about the horizontal and vertical widths. Glyphs have an origin point and left and right margins (sidebearings):

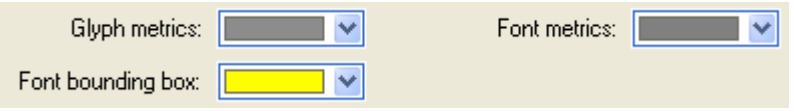


Left and **right margins** are used to define the positions of sequential glyphs in a series when the horizontal writing mode is selected. The **origin point** of a glyph is the cross point of the left margin and the font baseline.

The distance between the left and right margins is called the glyph's **advance width**.

The **ascender**, the **descender**, the **Caps height**, the **x height** and the **baseline** are font characteristics. Nonetheless you can edit these font properties in the Glyph window as well.

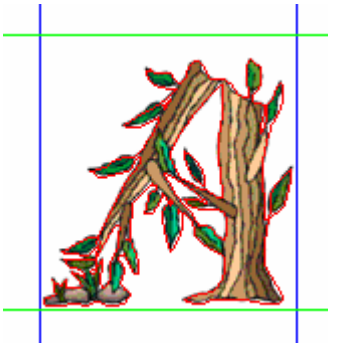
You may choose to change the lines colors and other viewing options in the Tools > Options dialog. Select the **Tools > Options** command in the **Tools** menu and refer to the **Glyph and Image Window** page described in the ***Glyph Window Options*** (on page 50) section:



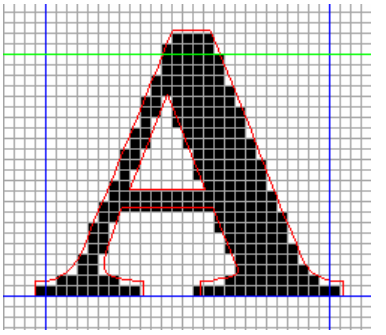
To edit glyph and font metrics, the *Edit Metrics* tool described later in the ***Editing Metrics in the Glyph Window*** (on page 308) section is used. To edit left and right margins position, the **Adjust Sidebearings** and **Adjust Width** operations in the Metrics window are used as well. Refer to the ***Using the Metrics Window*** (on page 309) section for details.

Outline Layer

Each glyph in a font has an *outline layer*. Usually it is empty because bitmap fonts do not contain outline information. But sometimes when working with the glyph's bitmap you may need to see its outline. An outline layer can be used in import/export operations only, but is never saved with the font (except in BitFonter Binary Font (.bfb) format). A Glyph with an outline contour may look like this (outline layer is in red on your screen):



or



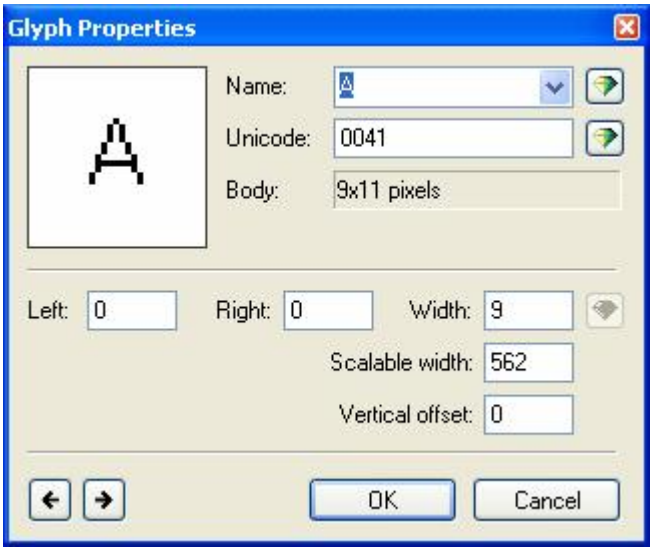
If the outline layer of a glyph is not empty, most operations that you make to the glyph are applied to its outlines too. Outlines are created automatically during the import operations (see the ***Importing a Font*** (on page 198) section), exporting to outline font (see the ***Exporting Outlines*** (on page 214) section) or manually (see the ***Outline Operations*** (on page 300) section later in the chapter).

Use the **Outline** command in the **View > Show Layers** menu to show or hide outlines in the Glyph window.

Refer to the **Glyph and Image Window** page of the Tools > Options dialog box to change the color and line thickness for contours in the Outline layer:

Editing Glyph Properties

To view or edit the glyph properties, select the **Properties** command in the **Edit** menu or press **Alt+Enter**. The Glyph Properties window will open:



You can view or edit the properties of all the existing glyphs in the font. Browse the glyphs using the **Previous** and **Next** buttons located at the left bottom of the window.

You can view and edit the following glyph attributes:

Name	The glyph's name from the standard name-Unicode database. Changing the name with the dropdown list to the right will change the Unicode codepoint automatically
Unicode	Unicode codepoint of the glyph, linked to the glyph's name through the standard name-Unicode database of BitFonter. Changing the Unicode codepoint will not change the name automatically
Body	Width and height of the glyph's shape in pixels. This is actually the glyph's bounding box which is not editable
Left Sidebearing	Horizontal offset of the glyph image from the glyph's origin point (left margin)
Right Sidebearing	Horizontal offset of the right margin from the glyph image (=Pixel Width — Left Sidebearing — body width)
Width	The advance width of the glyph, the distance between the left and the right margins of the glyph in pixels
Scalable width	Advance width in Units Per eM=1000 used for proper font scaling
Vertical offset	Vertical offset of the glyph image from the glyph origin point (or baseline).

Pressing the button at the right of the **Name** field generates glyph name from the Unicode codepoint below and vice versa: pressing the button at the right of the **Unicode** field generates Unicode codepoint from the glyph's name.

After you have finished editing click on the **OK** button or browse to the next/previous glyph. To discard changes to the current glyph click on **Cancel**.

Changing the View in the Glyph Window

To change the view in the editing field of the Glyph window, use the Scroll bars, **Size** dropdown list, **Zoom** buttons, and the **View** menu.

Browsing

The **Next** and **Previous** commands in the **Glyph** menu show the next and the previous glyphs of the font. You do not have to close one glyph to open another. You can browse all the existing font glyphs by using these commands in one open Glyph window. There are keyboard shortcuts for these commands: **Ctrl-[** and **Ctrl-]**.

Another way to get the glyph you need in the Glyph window is to use the alphanumeric keys of your keyboard. Pressing them brings up the corresponding glyph for edit if the button at the bottom of the Glyph window was not pressed. If it is pressed (locked), then pressing keys changes the current tool in the Tools toolbar.

Scrolling

You can scroll the editing field using the Scroll Bars. The same action can be done by the Hand tool, which becomes available when you hold down the **Spacebar** key.

The top and left rulers can be switched on and off by using the **Rulers** command in the **View** menu.

Zooming

You can define magnification or reduction of the editing field of a glyph with the **Size** dropdown list. There are 26 zoom modes available in the menu. The **Zoom-in** and **Zoom-out** buttons allow you to change the zoom mode of the editing field by one level at a time.

The maximum zoom-in is limited to 1600% magnification, and the minimum reduction is 10%. To magnify or reduce part of a glyph, select the Zoom tool (described ***later*** (on page 268) in this chapter).

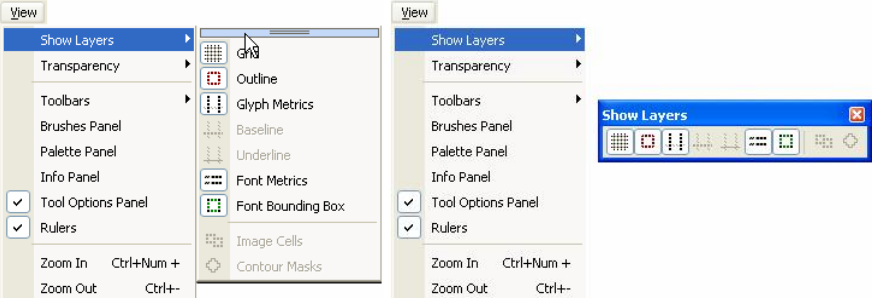
The physical size of the editing field is set to one and a half the size of the font bounding box by default. When you edit the glyph you are limited by the size of the editing field. To make the editing field larger, draw outside the bounding box, close and then open the glyph window once more. The editing field will fit the size of the new bounding box. You can set the fixed size of the editing field in the **Canvas and Dimensions** page of the Tools > Options dialog.

View Menu

In BitFonter every glyph contains several editing layers. Some of them are used when the font is exported; others are BitFonter-only and are used to help you work with the glyph. Below is the list of all the layers that you can see in the Glyph and Image window.

	Bitmap	Main layer containing the glyph's bitmap image
	Grid	Shows regular grid in the editing field in large zoom modes
	Outline	Shows glyph outline layer (it is usually empty)
	Glyph metrics	Shows glyph metrics — left and right sidebearings
	Baseline	Shows baseline
	Font Metrics	Shows vertical font metrics, such as ascender, descender or cap height
	Font Bounding box	Shows font bounding box rectangle in the editing field of the glyph window
	Image Cells	Shows cell rectangles in a split image (Image window)
	Contour Masks	Shows cell contours in a split image (Image window)

You can control the layers' appearance and features with the **View > Show Layers** menu. If you frequently need to switch some layers on and off you can drag the **View > Show Layers** menu and convert it to a toolbar:



The View > Transparency menu contains the following options:

As small grid	Shows glyph transparency as small grid
As medium grid	Shows glyph transparency as medium grid
As large grid	Shows glyph transparency as large grid
As color	Shows glyph transparency as color background
Select color	Allows you to choose color for transparency.

Choose **Transparency > As grid** if you want to see a transparent background in the Glyph window. Select the size of the grid from the Small grid, Medium grid, or Large grid choices in the **View > Transparency** menu. The size of the grid affects only the view of the editing field, not the glyph itself.

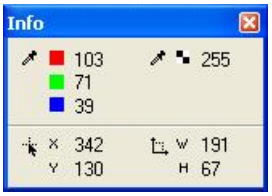
To view the glyph background as colored, select the **Transparency > As color** command in the **View** menu. The default color for transparency is white, but you can change it using the **Select Color** command (which brings up the standard Windows Select Color dialog) in the **View > Transparency** menu. Changing the view of transparency does not affect the real transparency of drawing colors. It defines only how the transparency is shown in the glyph window.

You may choose to change other viewing options on the **Glyph and Image Window** page of the Tools > Options dialog box described in the ***BitFonter Options*** (on page 44) section.

With the **View** menu, you can also show and hide toolbars, rulers and panels.

Info Panel

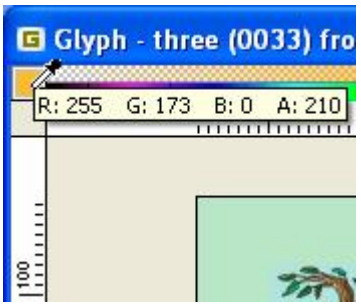
When in the Glyph or Image window, the Info panel may become useful. To make it visible on the screen, select the **Info Panel** command in the **View** menu:



The upper part of this panel contains color information for a pixel under the mouse cursor: its RGB values and the opacity value (0-255). If you are editing a font that has no transparency layer (less than with millions of colors), the opacity value is replaced with the gray level in percents. When black & white font is edited, RGB values are not shown:

The bottom part of the Info panel contains the horizontal (x) and vertical (y) coordinates of a pixel under the mouse cursor. The glyph's origin point has zero values. If you are in the Image window, the pixels are counted from the top left corner of the image. The next pair of values show the width (w) and the height (h) of a selection, if any.

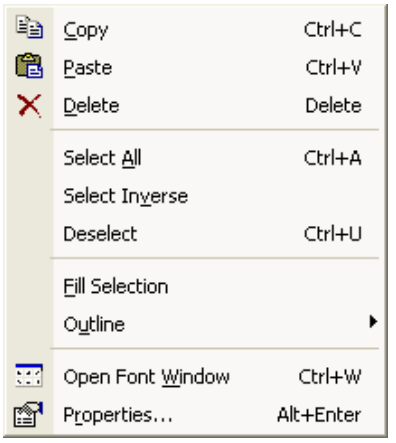
There is another way to open the Info panel: position the mouse cursor over the Current Color box in the Glyph or Image window and hold down the left mouse button:



The information panel for the current color appears. Drag the mouse outside this panel and release the left mouse button. The Info panel will appear on the screen where the mouse button was released.

Context Menu

Some commands available in the Glyph window can be selected from the context menu. To open the context menu, right-click anywhere in the editing field. Here is a sample of the Glyph window context menu:



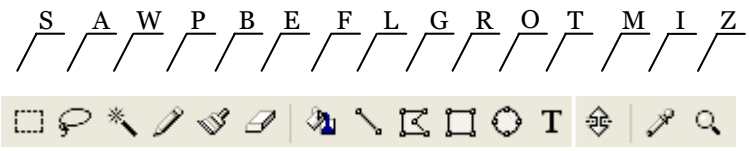
Here is what the commands mean:

Copy	Copies the selection onto the Clipboard. Same as the **Copy** command from the **Edit** menu
Paste	Places pixels from the Clipboard into the glyph image. Same as the **Paste** command from the **Edit** menu
Delete	Deletes the selection. Same as the **Delete** command from the **Edit** menu
Select All	Selects the entire editing field of the Glyph window
Select Inverse	Deselects the selected area and selects the unselected area simultaneously
Deselect	Deselects any previously selected area
Fill Selection	Fills the selected area with the current color
Outline	The commands from the **Tools > Outline** menu. They are described in the ***Outline Operations*** (on page 300) section later in the chapter
Open Font Window	Switches to the corresponding Font window
Properties	Opens the glyph Properties panel for the current glyph.

Tools

The tools in the Tools toolbar are used for glyph shape and metrics editing and are not available if the Glyph or Image window is not active. These tools are not used for outline layer editing. To open the Tools toolbar, select the **View > Toolbars > Tools** command.

You can easily switch between these tools by pressing the following keys on the keyboard:



These keyboard shortcuts will only work if the Direct glyph navigation button at the bottom of the Glyph window is in the *locked* position. If the button is in the *unlocked* position, pressing a keystroke on the keyboard will open the glyph for the character associated with that keystroke for editing in the current Glyph window.

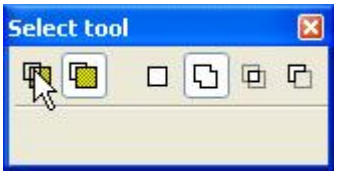
Selecting

You can select the entire editing field of the Glyph window by the **Select All** command in the **Edit** menu. If you want to deselect any previously selected area, use the **Deselect** command in the **Edit** menu. To deselect the selected area and select the unselected area simultaneously, use the **Select Inverse** command.

Additionally there are three selecting tools in the Tools toolbar: **Select** tool, **Select Region** tool and **Magic Wand**.

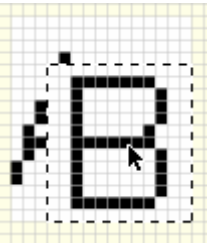
Any selected part of the glyph can be moved around the editing field by dragging it. It can be cut, copied, pasted, and cleared by the corresponding choices in the **Edit** menu or buttons in the Standard toolbar.

All the selecting tools have common options, which you can see in the Tool Options panel (the **View** menu). They are the *Transparent* selection and the *Opaque* selection:



You may need to switch between these options to make the selected part of a glyph transparent or opaque when it is moved around the editing field. Click on the **Transparent** button in the Tool Options panel if you want the white area of the selection *not* to cover the underlying black dots when the selection moves:

Click on the **Opaque** button if you want the white area of the selection to overlap the black areas of the underlying image:



You can change these options at any time, not just before making a selection.

The following selecting mode buttons

allow you to modify the selected area, i.e. to add or to subtract from the selection.

Any selection action can be undone by the **Undo** command of the **Edit** menu.

Select Tool

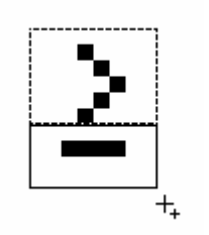
Use the Select tool to outline a rectangular area and display it with a grid.

To select a rectangular part of the glyph:

1. Choose the Select tool in the Tools toolbar.
2. Put the mouse cursor on one of the corners of the rectangle that you want to select.
3. Hold down the left mouse button and move the mouse cursor to the opposite corner of the area that you want to select.
4. Release the left mouse button.

➲ *Tip*: To temporarily switch to the Select tool while another tool is active, hold down the **Ctrl** key on the keyboard.

To add to the selection, hold down the **Shift** key or use the **Add to selection** button in the Tool Options panel. The cursor will change to . The new selection will be added to the previously selected area.



To subtract from the selection, hold down the **Alt** key or use the **Subtract from selection** button in the Tool Options panel. The cursor will change to . The new selection will be subtracted from the previously selected area.

To subtract from the selection, hold down **Alt+Shift** or use the **Intersect with selection** button in the Tool Options panel. The cursor will change to . Only the common area of the new and previous selection will be selected.

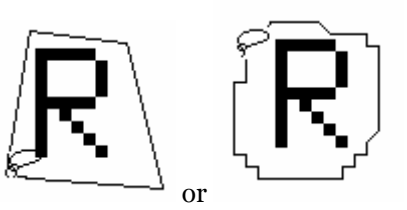
To deselect the area, click outside of the selected area, or **Ctrl**-click inside the area, or use the **Deselect** command in the **Edit** menu.

Select Region Tool

With the Select Region tool, you can select areas of any shape. The area may be selected on a point-by-point basis or by drawing a free-hand line.

To select a free-form part of the glyph:

1. Choose the Select Region tool in the Tools toolbar.
2. Position the mouse cursor on one of the points on the edge of the selection area.
3. Hold down or click the left mouse button. If you click the left down button, you will add a new straight line to the selection polygon. If you hold the left mouse button and drag the mouse, you will draw a freehand line that will be added to the selection:


or

4. Double-click the left mouse button to finish the selection.

To add to the selection, hold down the **SHIFT** key or use the **Add to selection** button in the Tool Options panel. The cursor will change to . The new selection will be added to the previously selected area.

To subtract from the selection, hold down the **ALT** key or use the **Subtract from selection** button in the Tool Options panel. The cursor will change to . The new selection will be subtracted from the previously selected area.

To subtract from the selection, hold down **ALT+SHIFT** or use the **Intersect with selection** button in the Tool Options panel. The cursor will change to . Only the common area of the new and previous selection will be selected.

To deselect the area, **CTRL**-click inside the area or use the **Deselect** command in the **Edit** menu.

Magic Wand Tool

The easiest selection tool is the Magic Wand. It automatically detects the area to select by its color attributes and creates a freehand selection. It is very useful in selecting a glyph shape.

To select a non-white area of the glyph:

1. Select the Magic Wand tool in the Tools toolbar.
2. Move the mouse cursor over the black area of the glyph that you want to select.
3. Click the left mouse button to select the glyph.

Hold down the **Shift** key to add a new selection to the current selection or hold down the **Alt** (or **Alt+Shift**) key to subtract from the selection.

To deselect the area, **Ctrl**-click inside the selected area or use the **Deselect** command in the **Edit** menu.

The Magic Wand tool has one additional option in the Tool Options panel:

It is called **Tolerance**. Its value can be changed in the range from 0 to 100. When tolerance is equal to 0, the Magic Wand will select pixels precisely of one color only. The greater the tolerance value is, the larger the area that will be selected by the Magic Wand. This option is, of course, not useful with black & white glyphs. All values except 100 will lead to selection of only the black pixels in this case.

Operations on a Selection

Selection is used to choose part of the glyph on which to apply transformations. In this section we will discuss methods of transforming selected areas.

Moving a Selection

To move a selected area:

1. Choose a selection tool (Select, Select Region or Magic Wand).
2. Move the mouse cursor inside the selection area.
3. Hold down the left mouse button and drag the selection to the new place.
4. Release the left mouse button to finish moving or press the **Esc** key on the keyboard to abort the operation.

To constrain movement to the horizontal or vertical direction, hold down the **Shift** key after pressing the left mouse button, and drag the mouse.

You can move the selected area by one pixel using the **arrow** keys.

Transparent and Opaque Modes

There are two drawing modes you can use when moving selections:

Opaque mode: you can move the entire selection, including black and white areas that work together, as an object. In this case the whole selection overlaps the underlying image.

Transparent mode: this only works on the black parts of the selection. Only the black areas in the selection will obscure the underlying image and the white areas will be completely transparent. With true color images all pixels overlap each other according to their transparency attributes.

Opaque mode *Transparent mode*

To select the Transparent or Opaque mode you must use the mode selection buttons in the **Tool Options panel**:

selects the Transparent mode

selects the Opaque mode

Copying and Pasting a Selection

To copy a selection into another application or into another glyph, use the **Copy** and **Paste** commands from the **Edit** menu.

With the **Copy** command you can place a copy of the selection area onto the Clipboard, making it available to any application that can read bitmap images.

The **Paste** command reads bitmap images from the Clipboard and places them into the current glyph's editing field. See also the ***Bitmap Pasting*** (on page 269) section.

The **Cut** command combines **Copy** and **Delete** operations, so a copy of the selection is placed on the Clipboard, but the selection is deleted.

Deleting a Selection

To remove a selection from the glyph image, select the **Delete** command in the **Edit** menu or press the **Del** key on the keyboard. The selection will be removed and filled with white or transparency color.

do not forget you can use the **Undo** command if you want to undo the deletion.

Transforming a Selection

You can use the commands from the **Tools** menu to transform a selected area of a glyph. If you have a selection within the glyph, transformation commands are applied to the selection. If there is no selection, the commands are applied to the entire glyph (or image in the Image window). All these operations will be discussed later in the ***Operations*** (on page 270) section.

Drawing

You can draw a new glyph or modify an existing one using the drawing tools. BitFonter's glyph editor includes the following drawing tools:

Pencil		Draws thin 1-pixel lines or sets or clears individual pixels. Very useful in the Zoom-In mode
Brush		Main drawing and painting tool. Draws strokes of selected width and shape
Eraser		Clears areas of a glyph
Fill		Fills closed areas with selected color
Line		Draws straight lines
Polygon		Draws filled or unfilled polygons
Rectangle		Draws filled or unfilled rectangles and squares
Ellipse		Draws filled or unfilled circles and ellipses
Text	T	Enters text

The default color for drawing is black. You can change the current color by selecting one in the Color Palette of the Glyph window. All the tools except the Text tool may draw with any color.

While drawing you can change your drawing tool temporarily:

- to the Select tool by holding down the **Ctrl** key, or
- to the Eye Dropper tool by holding down the **Alt** key, or
- to the Hand tool by holding down the **Spacebar** key, or
- to the Zoom-In tool by holding down the **Ctrl+Spacebar** keys, or
- to the Zoom-Out tool by holding down the **Ctrl+Alt+ Spacebar** keys.

➘ Note: If there is a selected area anywhere in the editing field, drawing will take place only within this selected area.

Selecting Colors

The glyph editor in BitFonter is actually a powerful color image-retouching module. As in any image editing application you can select drawing colors in color palettes. The glyph editor's color palette resides at the top of the Glyph window.

There are three ways to set the current color in the Glyph window:

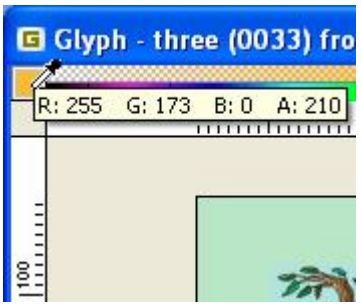
- With the Color Palette
- With the Color dialog box
- and with the Eyedropper tool

To set the current color with the Color Palette, move the mouse cursor to the palette bar (the cursor will change to the Eye Dropper cursor) and click the left mouse button over the color you need. The selected color will be shown in the Current Color box.

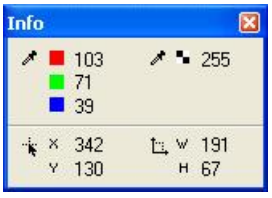
To set the current color with the Color dialog box, double-click on the palette bar or the Current Color box. The standard Windows Color dialog box will open. Select the color you need in this dialog and click on the **OK** button. The selected color will be shown in the Current Color box. This method is better if you want to set a precise color by entering its numeric RGB values.

To set the current color with the Eyedropper tool, select this tool in the Tools toolbar or press the **Alt** key on the keyboard; position the mouse cursor over the glyph image; and click the left mouse button. The color of the pixel you clicked on will be shown in the Current Color box.

To see the numeric values of the currently selected color, position the mouse cursor over the Current Color box. You will see the panel popups with RGB and opacity color information:



Hold down the left mouse button, drag the mouse and release the left mouse button to open the Info panel instantly:

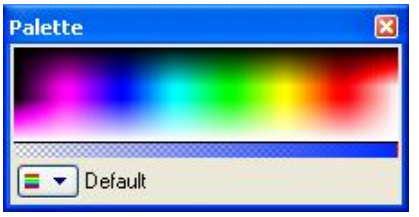


Note that the Info panel shows the color information for the image pixel under the mouse cursor, not the current color.

You can hide/show the Info Panel at any time by the command in the **View** menu.

Color Palettes

Sometimes it is useful to have the color palette in a separate floating window. To open this window, select the **Palette Panel** command in the **View** menu. The Palette panel will appear with the default color palette:

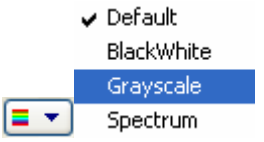


➲ Tip: You can drag this window out right from the Color Palette bar of the Glyph window.

Now you can use the Palette panel along with the Color Palette in the Glyph window. You can drag it around the screen to find the most convenient place. To resize the panel, click on and drag its right bottom corner.

There are two small controls at the bottom left corner of the Palette panel that are used to change the palettes in the window.

To change the color palette in the panel, use the **Select Palette** dropdown list:

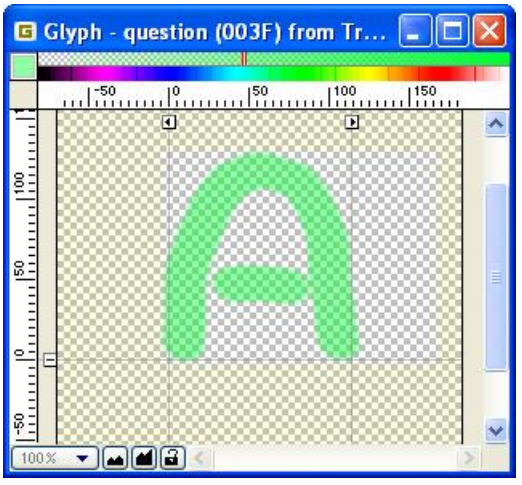


To view and use the standard color palette, click on the button if available. The Palette panel view changes:

The button is not available when thousands or millions of colors are used. The contents of the Palette panel when the button is pressed depend on the current font color mode.

Glyph Transparency

You can create fonts and draw glyphs with transparent colors. Transparency mode is available in true color (32-bit) fonts only. When you open a 32-bit Glyph window there is an element in it not found in other modes - the Transparency Palette. This new palette resides above the Color Palette at the top of the Glyph window:



The Transparency Palette can be used along with the Color Palette. You can think of transparency (opacity) as an additional color attribute. Every pixel in a glyph can be more or less transparent.

To set transparency for the current color, move the mouse cursor to the Transparency Palette (the cursor will change to the Eye Dropper) and click the left mouse button on the amount of transparency (opacity) you want. Or hold the left mouse button and drag the cursor left and right until the transparency you need is selected and then release the left mouse button. The selected transparency will be marked in the palette with a small black line or rectangle. If the Info panel is open, you can see the numeric values of different pixels' opacity. Zero means the pixel is fully transparent, 255 means it is fully opaque.

The transparency mode can be applied to all drawing tools used for glyph editing. Even the Eraser tool can erase pixels "partially".

➲ Tip: The Text tool can draw with white color in this mode.

Pencil Tool

The Pencil is the simplest drawing tool in BitFonter. It draws a one-pixel-wide line and can be used to set or clear individual pixels. The Pencil tool can be used at any zoom but it is easier to use it at zoom levels of 5:1 or greater.

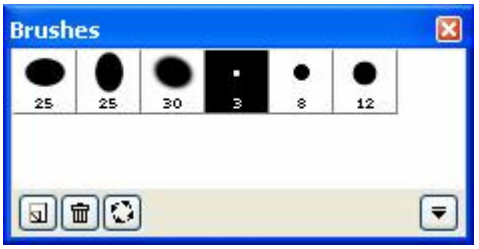
To set or clear an individual pixel on a glyph, move the mouse cursor onto the pixel you want to edit and click on the mouse button. This sets a pixel to the current color. Another click on the same pixel clears the pixel.

To draw a thin line with the current color, press the mouse button and drag the mouse. You can draw a straight line by clicking on a beginning node and then clicking on sequential final nodes while holding down the **Shift** key.

Press the **Esc** key on the keyboard while drawing to abort drawing a line.

Brush Tool

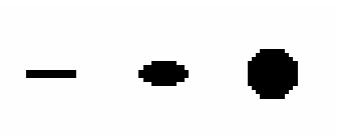
The Brush is a wide version of pencil. You select the brush shape in the Brushes panel:



Diameter is the brush shape diameter in pixels. The greater the value the wider the brush shape will be.

Hardness is the degree of diffusion of the brush edges. The greater the value the more diffused the brush edges will be. This option obviously does nothing in a plain black & white glyph.

Roundness is the form of the brush shape. If the roundness is zero, the brush is rectangular. The higher the value the rounder the brush. Here are examples of brushes of different roundness:



Angle defines the angle of the brush tip. It is applicable only to non-round brushes. If the angle is zero, the brush shape is horizontal. A positive angle turns the brush shape clockwise. A negative value turns the brush shape counter-clockwise. Here are examples of different brush angles:

Changing the **Roundness** and the **Angle** you can get calligraphic brushes of different shapes.

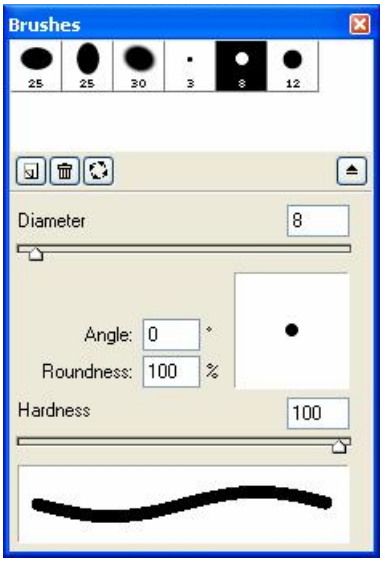
To use the Brush tool:

1. Select the Brush tool in the Tools toolbar and a color in the palette.
2. Set the brush diameter, roundness, angle and hardness in the Brushes panel.
3. Draw brush strokes by clicking on and dragging the mouse.
4. If you want to cancel a stroke while drawing, press the **Esc** key while dragging the mouse.

➲ Tip: To draw a straight line with the Brush, click once at the line beginning and then **Shift**-click on at the line end.

Brushes Panel

You can define several different brushes and place them in the Brushes panel. To make the Brushes panel visible, select the **Brushes Panel** command in the **View** menu:



To add a new brush to the top of the Brushes panel, define the brush's shape at the bottom and click on the button in the panel. The new brush will be added to the brushes list.

To remove a brush from the Brushes panel, click on the brush in the list to select it and then click on the button in the panel. The brush shape will be deleted from the brushes list.

To change the shape of a brush in the Brushes panel, click on the brush in the list to select it; define the brush's shape at the bottom; and then click on the button in the panel. The new parameters will be applied to the currently selected brush.

Tablet Options

If you have a digitizing tablet connected to your PC, you may need to set up the tablet options. Click on the Tablet button in the Tool Options panel to bring up the following Tablet Options dialog:

You can customize the behavior of your digital pen in this dialog. Check **Size**, **Hardness** and **Opacity** options in an appropriate combination to define the results of pressing the digital pen.

Eraser Tool

The Eraser tool may be used to clear portions of the image.

As with the brush tool, you may set diameter, roundness, angle and hardness of the eraser in the Brushes panel. Tablet options affect the Eraser tool as well.

Of course, you can press the **Esc** key to abort the erasing process.

Fill Tool

The Fill tool is used to fill areas filled with one color. It works similarly to the Magic Wand tool.

To fill an area (filled with one color) with the current color using the Fill tool:

1. Select the Fill tool in the Tools toolbar and a color in the palette.
2. Move the mouse cursor to the area you wish to fill.
3. Click the left mouse button to fill this area with the current color.

To fill a *selection* with the current color, use the **Edit > Fill Selection** command.

Line Tool

This tool draws straight lines with the brush that you select in the Brushes panel.

To draw a straight line:

1. Select the Line tool in the toolbar and a color in the palette.
2. Position the mouse cursor + at the beginning of the line.
3. Hold down the left mouse button and drag the mouse to define a line.
4. Hold down the **Shift** key to constrain the line direction to 45-degree increments.
5. Release the left mouse button to finish drawing the line or press the **Esc** key to abort.

Polygon Tool

This tool can be used to draw closed polygons with the interior filled or empty.

As with all the other tools you can choose the current color in the Color Palette.

To draw a polygon:

1. Select the Polygon tool in the Tools toolbar.
2. Position the mouse cursor + on the first point of the polygon.
3. To add a straight-line segment to a polygon, click the left mouse button. Then move the mouse to define the line segment and click the left mouse button again.
4. To add a freehand segment to a polygon, press down the left mouse button (but do not release it) and drag the mouse to draw the freehand segment. Release the left mouse button to accept it.
5. To finish drawing a polygon, double-click the left mouse button at the last point to be added.
6. To abort drawing, press the **Esc** key.

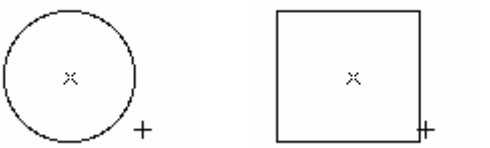
The polygon will be filled with the current color if the **Fill interior** check box in the Tool Options panel was checked:

Rectangle and Ellipse Tools

These two tools are very similar in usage, but differ in result. The Rectangle tool is used to draw rectangles and squares, and the Ellipse tool draws ellipses and circles.

To draw a rectangle or an ellipse:

1. Select the appropriate tool, or , in the Tools toolbar and a color in the palette.
2. Position the mouse cursor + on one of the rectangle's corners. If you are drawing an ellipse, position the cursor on one of the corners of the imaginary rectangle that surrounds the ellipse.
3. Hold down the left mouse button and drag the mouse to draw a rectangle or an ellipse.
4. Hold down the **Shift** key while drawing to draw a square instead of a rectangle, or a circle instead of an ellipse.
5. Hold down the **Alt** key while dragging to place the center of the ellipse or rectangle at the place where you initially positioned the mouse cursor:



or

   Pressing the **Alt** key before you press the mouse button will change the tool temporarily to the Eye Dropper tool.
6. Release the left mouse button to finish drawing or press the **Esc** key to abort.

The ellipse or rectangle will be filled with the current color if the **Fill interior** check box in the Tool Options panel is checked. To temporarily change the fill mode, hold down the **Ctrl** key while dragging.

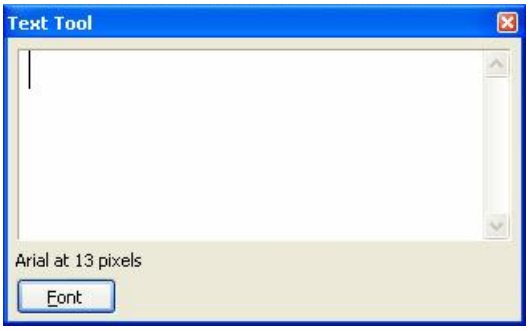
Text Tool

With this tool (which is a little more complex than the other drawing tools) you can draw letters, words or paragraphs of text using any font installed on your system. This feature is very useful, because it allows you to use any font as an original for further modifications.

You can draw letters with any color from the Color Palette except white.

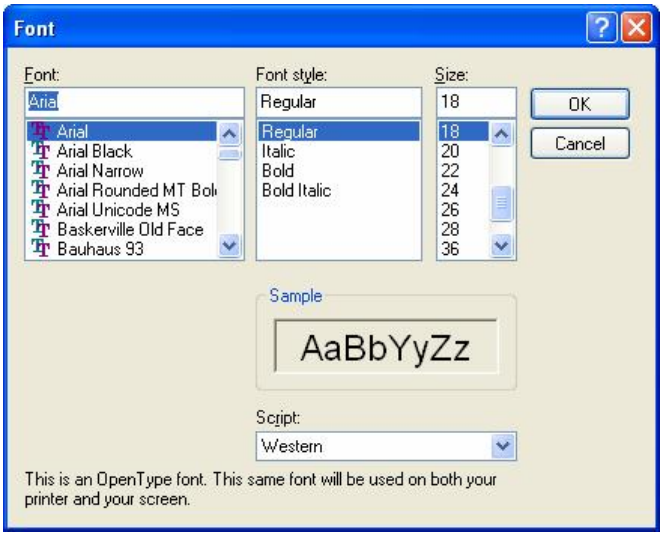
To draw a string of text:

1. Select the Text tool T in the Tools toolbar and click on in the glyph editing field.
2. You will see a panel for entering text:



In this panel you can enter a line of text and choose the font that will be used to draw the text.

3. Click on the **Font** button at the lower left corner to select the font. You will see the font-selection dialog box, in which you can select the typeface, style and point size of the text that you are drawing:



4. Make your selection and click on the **OK** button.

5. Type the letter or text that you want to draw in the glyph window. You will see the text objects containing the text immediately appear in the glyph window while you are typing:

6. You can work with the text object as with any selection. Use the Text tool like any other selection tool to move or copy a text object. Refer to the ***Selecting*** (on page 244) section for information about selection tools.

7. Choose a transparent or opaque mode for the text object in the Tool Options panel.

8. Click the left mouse button anywhere outside the selection to accept the position of the text object and begin entering a new one.

Other Tools

Eye Dropper Tool

The Eye Dropper is a tool for quick color selection from the colors already used in the glyph drawing. You can choose the Eye Dropper by selecting the tool in the Tools toolbar or by pressing and holding down the ALT key while another tool is active. To make any color in the editing field the current color, drag the lower edge of the mouse cursor to that color and click the left mouse button. The color under the cursor becomes the current color for drawing tools.

If the Info panel is visible, the digital RGB values of the current color appear in it.

You cannot undo this operation.

Edit Metrics Tool

This tool is used for glyph and font metrics editing. Glyph metrics include the left and right margins of the glyph. Font metrics that can be edited in the Glyph window are the baseline, font ascender, font descender, Caps height and x height.

To edit glyph metrics:

1. Select the Edit Metrics tool in the Tools toolbar.
2. Position the mouse cursor on one of the lines representing metrics in the editing field. The cursor will change to or depending on what line — horizontal or vertical — you position it on.
3. Hold down the left mouse button and drag the line to its new place.
4. Release the left mouse button.

➘ Note: Dragging the baseline does not affect font metrics. It moves the glyph's shape in the reverse direction, adjusting its vertical position in the font.

To edit font metrics:

1. Select the Edit Metrics tool in the Tools toolbar.
2. Position the mouse cursor on any font metrics line in the editing field. The cursor will change to .
3. Hold down the left mouse button and drag the line to the new place.
4. Release the left mouse button. Font metrics will change, affecting all the font glyphs.

Holding down the **SHIFT** key while the cursor is positioned over a metrics line (which changes the cursor to or) and clicking on the mouse will move the line to the edge of the glyph shape. This works with both glyph and font metrics.

➲ Tip: To edit glyph and font metrics without selecting the Edit Metrics tool, you can use controls represented by signed color rectangles next to each metrics line.

➲ Tip: You can also change glyph metrics with the help of the **Width** and **Sidebearings** commands in the **Glyph > Metrics** menu.

➘ Dragging font metrics lines does affect font metrics i.e. all the glyphs in the font.

Zoom Tool

Often the size of the glyph that you are editing is smaller than the size of the window. You may need to see more details of an image. Do this by using the **Zoom-in** and **Zoom-out** buttons in the Glyph window or the Zoom tool.

To change the zoom mode:

1. Select the Zoom tool in the Tools toolbar.

2. Position the mouse cursor (or by holding down the **Alt** key) on one of the corners of the glyph region that you want to zoom in on (or zoom out).

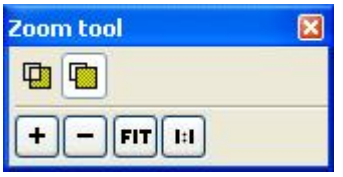
3. Hold down the left mouse button and move the mouse to select the region.

4. Release the left mouse button. The new zoom mode will be selected.

To change the zoom mode by one level, simply click on (or **Alt**-click) somewhere in the editing field.

To get the Zoom tool temporarily while another tool is selected, hold down **Ctrl+Spacebar** or **Ctrl+Alt+Spacebar**.

If the Zoom tool is selected the Tool Options panel includes some additional zoom buttons:



+ increases the zoom mode by one level

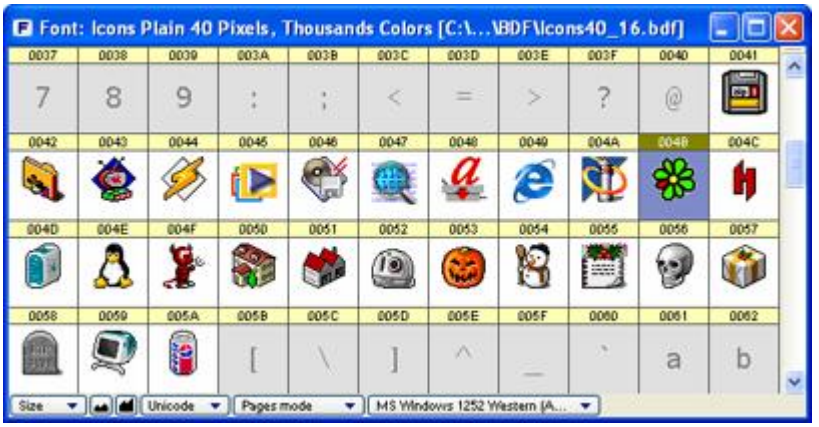
− decreases the zoom mode by one level

FIT fits the editing field to the size of the Glyph window

1:1 sets the view to 100%.

Bitmap Pasting

While the Glyph window is open you can use the Clipboard for pasting graphics into your glyph. Place the image onto the Clipboard in advance. A wide range of possibilities becomes available. Imagine a font that consists of your favorite icons, for example. You can build it in a few minutes by copy-pasting icon images from your favorite image manager:



Common bitmap pasting (using the **Paste** command in the **Edit** menu) can be performed only to the Glyph window. Drag-and-drop pasting can be performed by dragging a selection from other applications supporting drag-and-drop techniques to BitFonter's Font window. The pasting and drag-dropping processes are fully customizable. Refer to the **Bitmap Pasting** page of the Tools > Options dialog discussed in the ***BitFonter Options*** (on page 44) section.

Operations

You can transform a set of glyphs, one glyph, or a selected part of a glyph using the commands from the **Tools** menu. The **Tools** menu is also available when working with scanned or imported bitmap images. What the commands are applied to depends on what window is active and what is selected at that moment. If the Glyph window is active and there is no selection, transformation commands are applied to the particular glyph. If you select a part of a glyph or an image, transformations are applied to the selection. To transform a set of glyphs or an entire font, make the Font window active, select the glyph cells for transformation and choose the appropriate command in the **Tools** menu.

➘ Note: Some commands are not available in the **Tools** menu if you are editing a black & white font or image.

Some commonly used transformation commands have corresponding buttons in the Operations toolbar for easy access:

You can turn the toolbar on or off with the help of the **View > Toolbars** menu.

The buttons in the Operations toolbar are:

Flip Horizontal		Flips the glyph(s) or selection horizontally
Flip Vertical		Flips the glyph(s) or selection vertically
Scale		Scales the glyph(s) or selection
Rotate		Rotates the glyph(s) or selection by the given angle
Rotate 180°		Rotates the glyph(s) or selection 180 degrees
Rotate 90° CW		Rotates the glyph(s) or selection 90 degrees clockwise
Rotate 90° CCW		Rotates the glyph(s) or selection 90 degrees counter-clockwise
Slant		Slants the glyph(s) or selection to imitate italics
Make Bold		Shifts the glyph(s) or selection by several pixels to imitate bold style

Do not forget that you can use the **Undo** command if you want to undo any of the transformations.

Flip Horizontal and Flip Vertical

When you want to flip a glyph in the vertical or horizontal direction, you use the **Flip Horizontal** or **Flip Vertical** commands from the **Tools** menu.

Before flipping

After horizontal flipping

After vertical flipping

As with other commands from the **Tools** menu this command may be applied to the selected areas of a glyph, but when there is no selection, it will apply to the entire glyph. In the latter case glyph outlines (if any) are flipped accordingly. These operations may be necessary when you edit symmetrical glyphs.

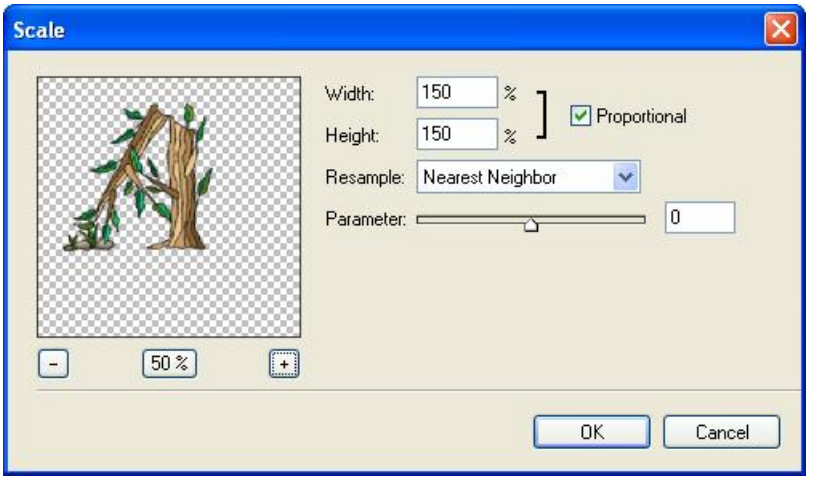
Use the **Undo** command to undo the transformation.

Scale

The **Scale** command allows you to resize the glyph or selected part of the glyph by adding or deleting the appropriate number of pixels. This function is useful for creating a larger font from a smaller one or vice versa. Also you can transform a normal font into a narrow one by reducing the width of its glyphs. But remember, this operation does nothing with the glyph's metrics. (See also the ***Changing Font Size/Resolution*** (on page 149) section in the ***Editing Fonts*** (on page 87) chapter.)

To scale the glyph:

1. Open the glyph's window.
2. Select the **Scale** command in the **Tools** menu. You will see the Scale dialog:



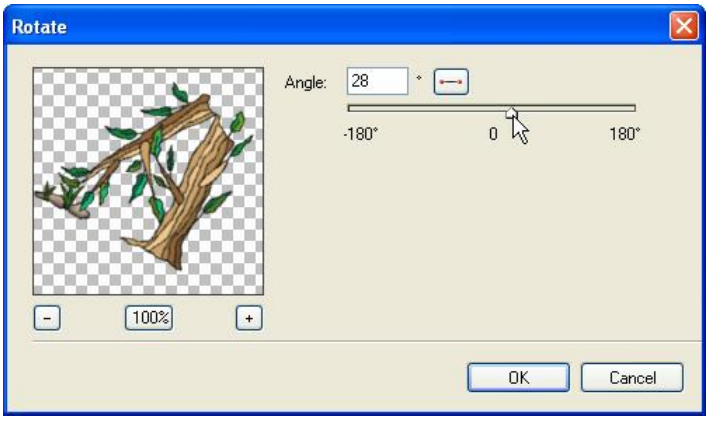
3. Enter the scale factor for horizontal and vertical transformation and select the scaling method in the **Resample** dropdown list. Check the **Proportional** check box if you want to keep the vertical and horizontal scale factors the same. Then click on the **OK** button. To abort scaling, click on **Cancel**.

The **Resample method** is the way scaling will be performed. When you choose a method other than *Simple*, the slider becomes available. Use it along with the method to apply different effects during the scale operation. It may take a little experimentation to find the best algorithm for your glyph image. In any case, you can edit the glyph manually after scaling.

You can see the preview of a glyph in the preview field of the dialog while changing the parameters of transformation. To see the original glyph, simply press and hold down the **Ctrl** key.

Rotate

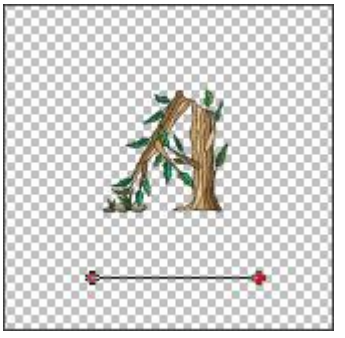
To rotate the glyph by a specific angle, select the **Tools > Rotate > Arbitrary** command. The Rotate dialog appears:



Enter the angle for rotation in the **Angle** text field or use the slider below. Positive values rotate the image clockwise.

You can see the preview of the glyph in the preview field of the dialog while changing the angle for rotation. To see the original glyph, hold down the **Ctrl** key.

The button allows you to set the rotation angle with the vector right in the preview field. Click on the button and draw a line in the preview box. For example, the horizontal line drawn from left to right corresponds to zero angle:



When finished click on the **OK** button. To abort the operation, click on **Cancel**.

Rotate 90°CW, Rotate 180° and Rotate 90°CCW

There are another three commands for glyph rotation in the **Tools > Rotate** menu. Use **Rotate 90˚CW**, **Rotate 180˚**, and **Rotate 90˚CCW** commands to rotate by the appropriate angle.

Before rotating

*After rotating 90°*CW

After rotating 180°

*After rotating 90°*CCW

If these commands are applied to the entire glyph or a set of glyphs outlines (if any) are rotated accordingly.

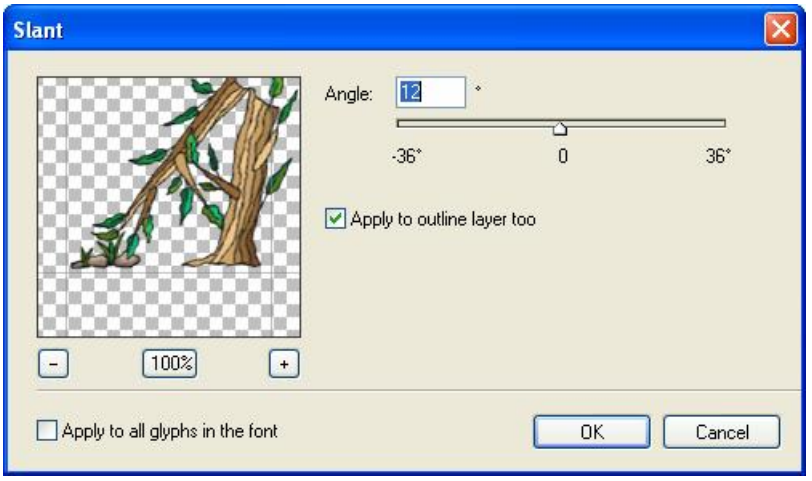
Use the **Undo** command, if you want to undo the transformation.

Slant

This is the special transformation command that can help you make an oblique font from a plain one.

To slant the glyph:

1. Select the **Slant** command in the **Tools** menu. You will see the Slant dialog:



2. Enter the angle for slanting (12 degrees by default) and click on the **OK** button. To abort the operation, click on **Cancel**.

Before

After

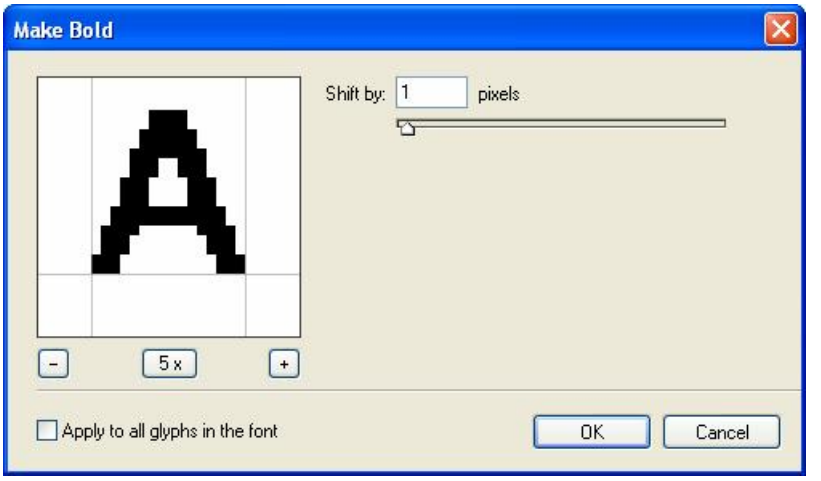
You can see the preview of the glyph in the preview field of the dialog while changing the slant angle. To see the original glyph, press and hold down the **Ctrl** key.

Make Bold

This is another special command that can help you build bold typefaces. It uses the scaling algorithm to make the glyph “fatter”.

To make the glyph bold:

1. Select the **Make Bold** command in the **Tools** menu. You will see the Make Bold dialog:



2. Enter the **Shift** value in glyph pixels and click on the **OK** button. The glyph will become “fatter”:

Before *After*

To see the original glyph in the preview field, press and hold down the **Ctrl** key.

Fine-tune the glyph using the drawing tools if needed.

Shift

The Shift operation is used to move the whole glyph's shape or its selected part in vertical or horizontal direction relatively to glyph's and font metrics.

When you select the **Shift** command in the **Tools** menu, the Shift dialog opens:

The only thing that you can do with this operation is to move the glyph(s) horizontally or vertically. For example, you may want to move several glyphs up or down at a time. Positive values shift glyph(s) up and right, negative values shift down and left.

You can use the **Undo** command to undo the Shift operation.

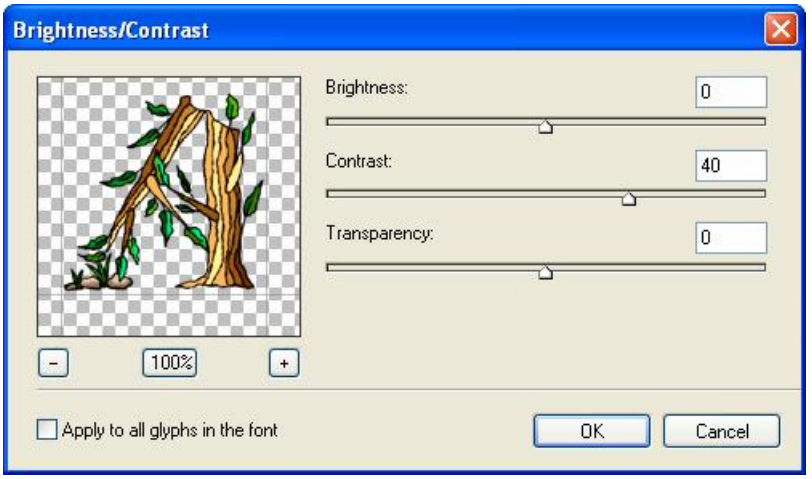
Clear Bitmap

The **Tools > Clear Bitmap** command erases the contents of the glyph bitmap layer. Outlines in the outline layer are preserved if any. Use the **Undo** command, if you want to undo the operation.

Brightness/Contrast

The next two operations are useful for retouching color glyphs. As seen from the command name you can change the brightness and the contrast of glyphs with this command. In addition this operation helps control glyph transparency.

To edit the glyph's brightness or contrast, select the **Brightness/Contrast** choice in the **Tools > Adjust** menu. You will see the dialog:



Move the **Brightness** slider to the left to make the glyph darker or to the right to make it lighter. Move the **Contrast** slider to the left to make the glyph lower contrast or to the right to make it higher contrast. Use the **Transparency** slider to make the glyph more or less transparent.

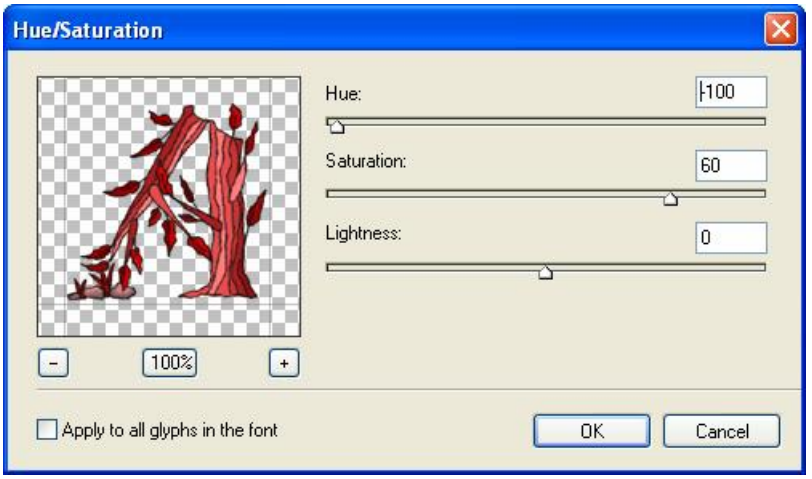
You can view the preview of the glyph in the preview field of the dialog. To see the original glyph, press and hold down the **Ctrl** key. Click on the **OK** button to apply the changes when finished, or **Cancel** to abort the operation.

➘ Note: This operation does not work with black & white fonts.

Hue/Saturation

This operation is also useful for retouching color glyphs. As seen from the command name you can use it to change the hue and the saturation of a glyph's colors.

To edit the glyph's hue, saturation or lightness, select the **Hue/Saturation** item in the **Tools > Adjust** menu. You will see the Hue/Saturation dialog:



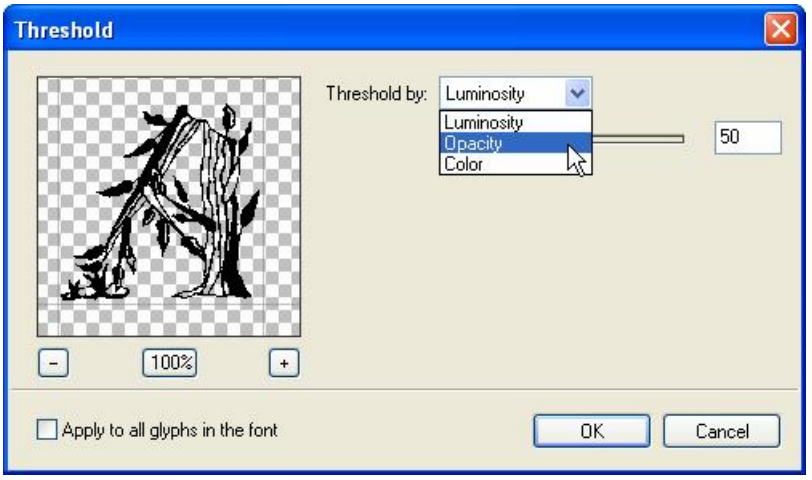
Use the **Hue** slider to change the color hue of the glyph. Move the **Saturation** slider to the left to make the colors less saturated or to the right to make them more saturated. Move the **Lightness** slider to the left to make the glyph darker or to the right to make it lighter.

You can preview the glyph in the preview field of the dialog. To see the original glyph, press and hold down the **Ctrl** key. Click on the **OK** button to apply changes when finished, or **Cancel** to abort the operation.

➘ Note: This operation does not work with black & white fonts.

Threshold

The **Threshold** command fills light pixels of the glyph image with white color and dark pixels with black. Actually it turns color glyphs into black & white glyphs, but it does not change the font's color mode. The boundary between "light" and "dark" can be defined in the Threshold dialog:



The threshold operation can be performed by one of the following parameters:

Luminosity	The luminance countered from pixel's red, green and blue channels
Opacity	Pixel's opacity/transparency component (for fonts with millions of colors)
Color	Pixel's hue angle

To see the original glyph, press and hold down the **Ctrl** key.

Before

After

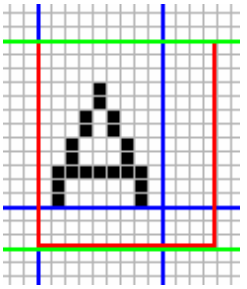
This command cannot be applied to a glyph more than once.

➘ Note: This operation does not work with black & white fonts.

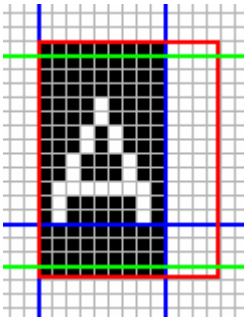
Invert

The **Invert** command inverts glyph pixels' colors. Only the selected part of the open glyph or glyphs in the Font window is inverted. You can invert the whole glyph in the Glyph window by selecting it with any selection tool.

In the simplest case the white pixels become black and the black pixels that compose the glyph shape become white. The surrounding background pixels become black. The right border of the background is the glyph's right margin. The left border of the background is the glyph's left margin or the left side of the bounding box. Vertically the background is limited by the bounding box, which becomes 2 pixels higher:

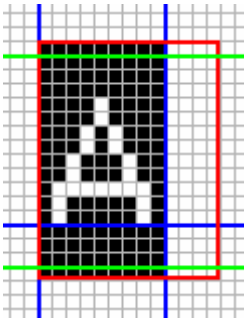


Before

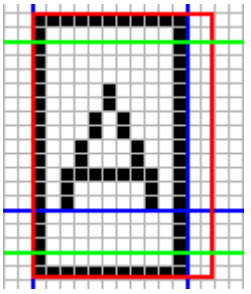


After

Inverting a second time does not return to the original glyph. Instead it becomes 2 pixels wider and 2 pixels higher:



Before



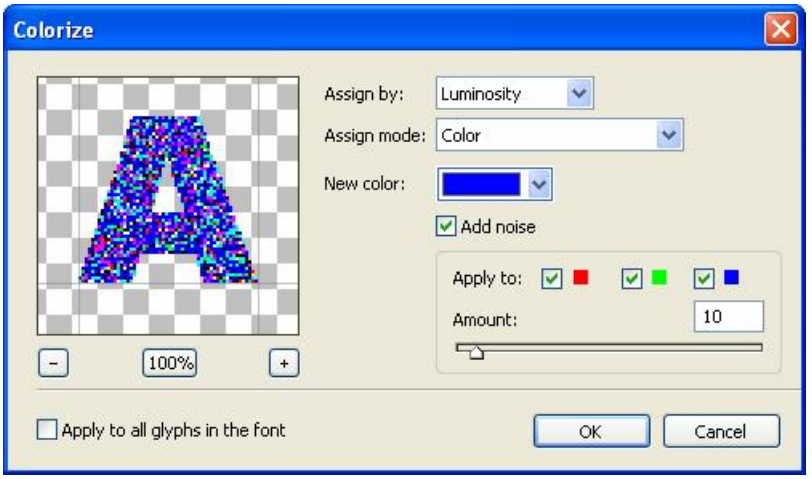
After

The red rectangle in the samples above represents the font bounding box. Use the **Undo** command to undo the operation.

Colorize

The **Colorize** command is the opposite of the **Threshold** command in a sense. It does not change the font's color mode either, but it allows you to fill all black & white glyphs with one color or even with a color/grayscale pattern.

To colorize the glyphs, select the **Colorize** command in the **Tools > Filter** menu and select the options in the Colorize dialog:

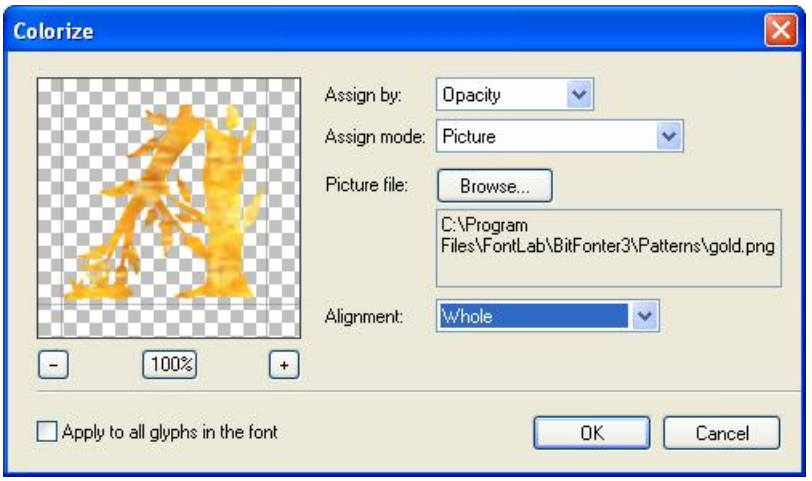


Select a method of determining what pixels should be colorized in the **Assign by** dropdown list.

Select a mode of glyph colorizing: color or picture.

If you choose **Color**, select the color that will be applied to the glyph's shape in the standard Colors dialog. Check the **Add Noise** option to add some noise to a solid color you have selected. You may choose to add different amount of noise and apply it to this or that component of RGB.

If you choose the **Picture** mode, click on the **Browse** button and choose the PNG or TIFF image file to be applied to the glyph's shape:



You may choose any image that can be produced with your image-editing applications, even your latest digital photo.

Select the appropriate **Alignment** in the corresponding menu. The following variants of aligning the image to the glyph's shape are available:

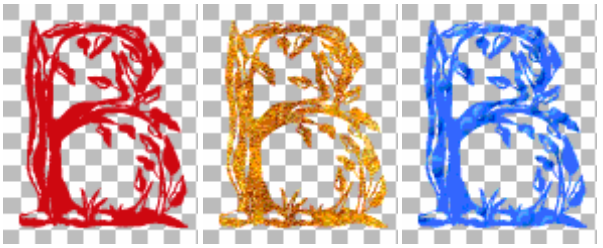
Random	The glyph's shape is filled with the image randomly. If the image size is larger than the glyph's shape, then different glyphs will be filled with different areas of one image
Whole	The glyph's shape is filled with the image scaled to the glyph's size. All the glyphs will get the whole image
Top Left	The image fills the glyph's shape aligned to the image's top left corner
Top Centered	The image fills the glyph's shape aligned to the image's top edge and centered horizontally
Top Right	The image fills the glyph's shape aligned to the image's top right corner
Centered Left	The image fills the glyph's shape aligned to the image's left edge and centered vertically
Centered	The glyph's shape is filled with the image centered vertically and horizontally in the glyph
Centered Right	The image fills the glyph's shape aligned to the image's right edge and centered vertically

Bottom Left	The image fills the glyph's shape aligned to the image's bottom left corner
Bottom Centered	The image fills the glyph's shape aligned to the image's bottom edge and centered horizontally
Bottom Right	The image fills the glyph's shape aligned to the image's bottom right corner.

View the preview of the glyph in the preview field of the dialog. To see the original glyph, press and hold down the **Ctrl** key. Click on the **OK** button to apply changes when finished, or **Cancel** to abort the operation.

Here are some examples of the Colorize command:

Before



After

Remove Background

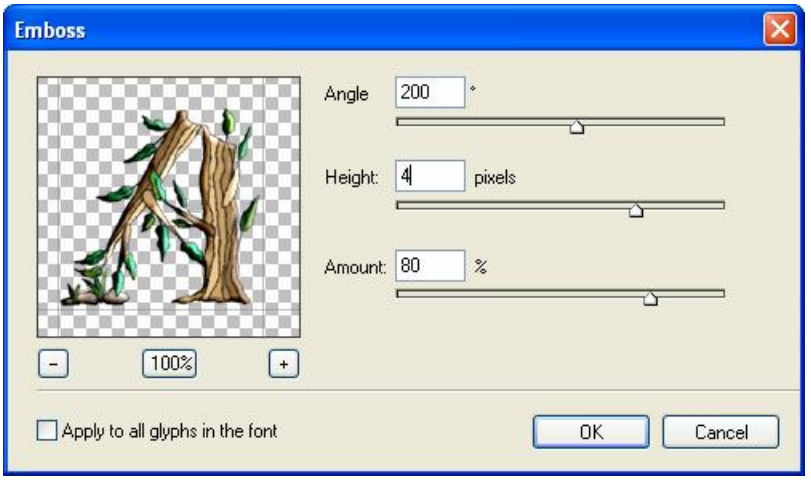
One more way to make the glyphs cleaner is to use the **Remove Background** command in the **Tools > Filter** menu. This filter tries to separate the glyph's shape from its background and then clear the background. We illustrated the use of this operation earlier when spoke about creating a font from an image in the ***Working with an Image*** (on page 172) section.

Emboss

This command and the remaining ones from the **Tools > Filter** menu may be qualified as filters. They are usually used for color glyph editing. **Emboss** is a special filter command that will make the glyph(s) look high relief.

To make the glyph high relief (embossed):

1. Select the **Emboss** command in the **Tools > Filter** menu. You will see the Emboss dialog:



2. Use the **Angle**, **Height** and **Amount** sliders to customize the filter and click on the **OK** button. The Emboss filter will be applied.

To see the original glyph in the preview field, press and hold down the **Ctrl** key.

Before

After

Smart Blur

The **Smart Blur** command blurs the glyph. It is similar to the **Gaussian Blur** command described later, but it tries *not* to blur the glyph's edges.

To perform this operation, select the **Smart Blur** command in the **Tools > Filter** menu. The Smart Blur dialog will appear:

Use the **Level** slider to define the blur degree.

You can see the preview of the glyph in the preview field of the dialog. To see the original glyph, press and hold down the **Ctrl** key. Click on the **OK** button to apply the transformation or **Cancel** to abort. You can use this command several times, but be careful to check the effect after every use.

Before

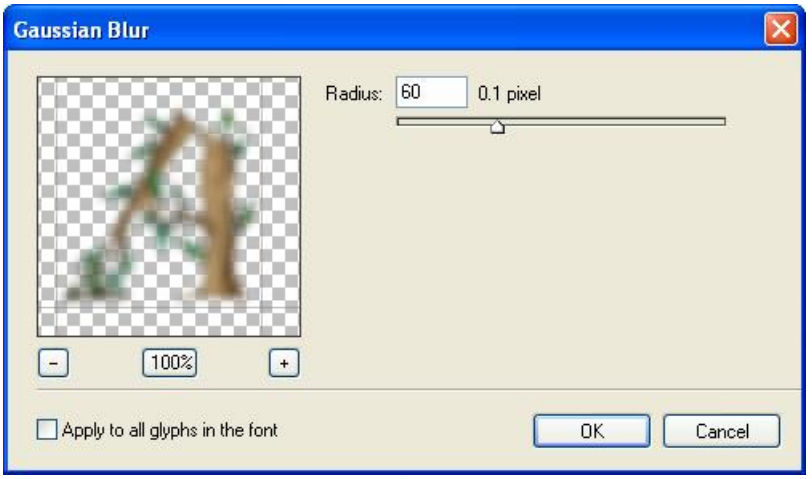
After

➘ Note: This operation does not work with black & white fonts.

Gaussian Blur

The **Gaussian Blur** command blurs the glyph using the Gaussian algorithm.

To perform this operation, select the **Gaussian Blur** command in the **Tools > Filter** menu. The Gaussian Blur dialog will appear:



Use the **Radius** slider to define the blur degree.

View the preview of the glyph in the preview field of the dialog. To see the original glyph, press and hold down the **Ctrl** key. Click on the **OK** button to apply the transformation or **Cancel** to abort. You may use this command several times, but be careful to check the effect after every use.

Before

After

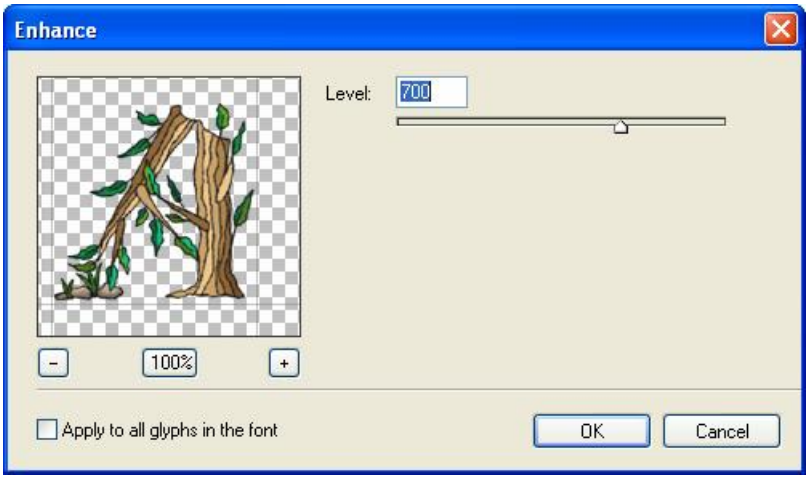
Use the **Undo** command to undo the operation.

➘ Note: This operation does not work with black & white fonts.

Enhance

The **Enhance** command tries to detect unwanted pixels and remove them. It is similar to the **Reduce Noise** command described later as it reduces noise, thus improving the glyph's appearance.

To enhance the glyph, select the **Enhance** command in the **Tools > Filter** menu. The Enhance dialog will appear:



Use the **Level** slider to define the degree of unwanted pixels to remove. This filter is good for scanned black & white fonts too. See also the examples of the ***Custom Filter*** (on page 297) usage described later.

You will see the preview of the glyph in the preview field of the dialog. To see the original glyph, press and hold down the **Ctrl** key. You can use this command several times, but be careful to check the effect after every use.

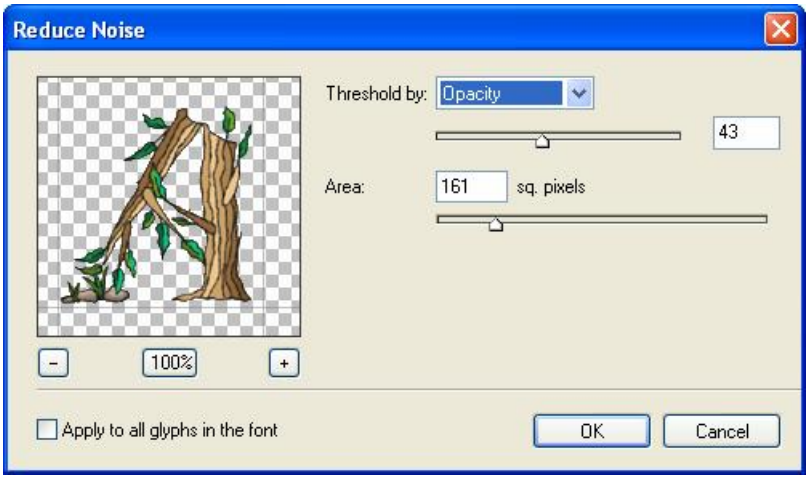
Before *After*

Use the **Undo** command to undo the operation.

Reduce Noise

The **Reduce Noise** operation makes the glyph less noisy.

To apply this operation, select the **Tools > Filter > Reduce Noise** command. The Reduce Noise dialog will appear:



Enter the **Threshold** method, amount and the **Area** value in square pixels to define the best transformation parameters. View the preview of the glyph in the preview field of the dialog. To see the original glyph, press and hold down the **Ctrl** key.

The **Threshold by** parameters work the same way as the ***Threshold*** (on page 282) operation.

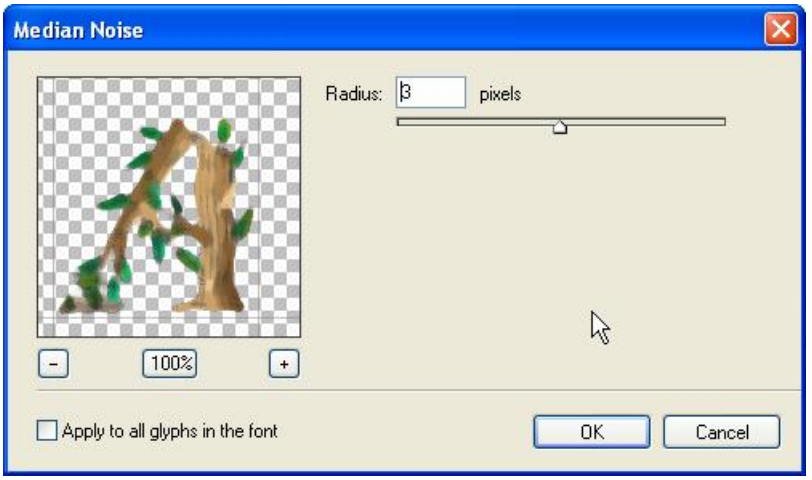
Click on the **OK** button to apply the transformation or **Cancel** to abort.

This operation can clean up the glyph image and even make it empty. You may use the **Reduce Noise** command several times to achieve the best result.

Median Noise

The **Median Noise** operation makes the glyph more or less noisy by averaging the colors of neighbor pixels.

To apply this operation, select the **Tools > Filter > Median Noise** command. The Median Noise dialog will appear:



Enter the **Radius** value for pixel color averaging. You will see the preview of the glyph in the preview field of the dialog. To see the original glyph, simply press and hold down the **Ctrl** key.

Click on the **OK** button to apply the transformation or **Cancel** to abort.

Before

After

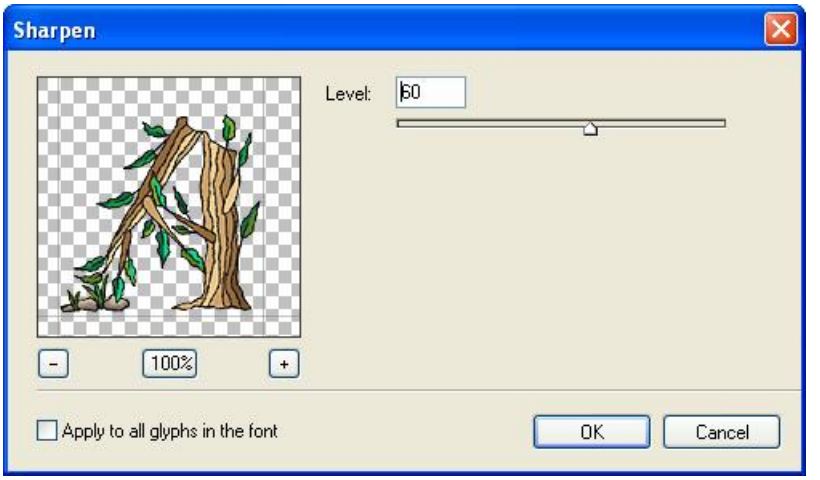
Use the **Undo** command to undo the operation.

You can use the **Median Noise** command multiple times to achieve the result you need.

Sharpen

This command makes the glyph image sharper and higher contrast. It is the opposite of the ***Soften*** (on page 296) command.

To apply this operation, select the **Tools > Filter > Sharpen** command. The Sharpen dialog will appear:



Enter the **Level** value of sharpness. View the preview of the glyph in the preview field of the dialog. To see the original glyph, press and hold down the **Ctrl** key.

Click on the **OK** button to apply the transformation or **Cancel** to abort.

Before

After

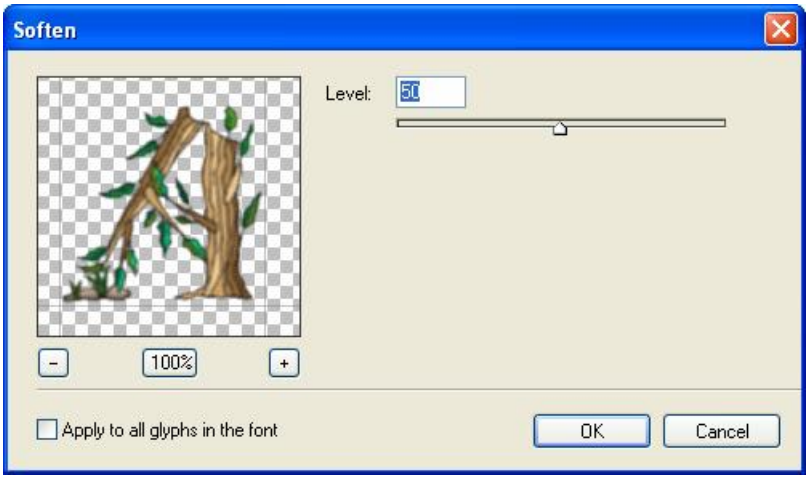
Use the **Undo** command to undo the operation.

You may use the **Sharpen** operation several times to achieve the best result.

Soften

This command makes the glyph image less sharp and lower contrast. It blurs the image. It is the opposite of the ***Sharpen*** (on page 295) command.

To apply this operation, select the **Tools > Filter > Soften** command. The Soften dialog will appear:



Enter the **Level** value of softness. View the preview of the glyph in the preview field of the dialog. To see the original glyph, press and hold down the **Ctrl** key.

Click on the **OK** button to apply the transformation or **Cancel** to abort.

Before

After

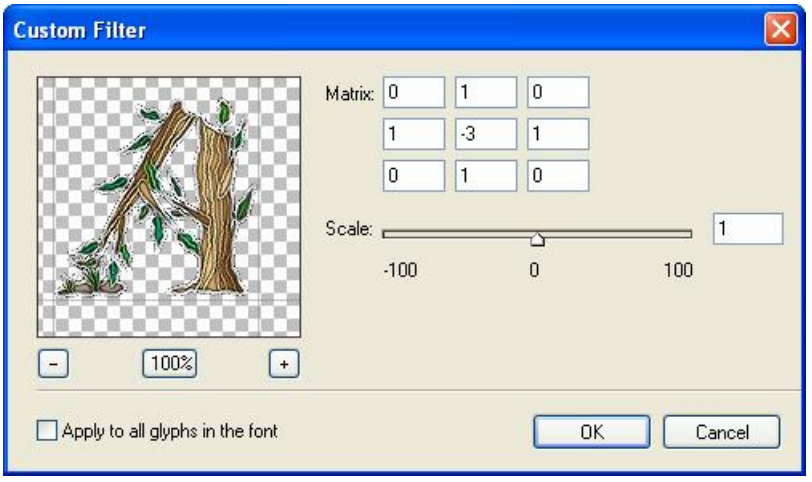
Use the **Undo** command to undo the operation.

You may use the **Soften** operation several times to achieve the best result.

Custom Filter

The **Custom Filter** command will allow you to make many transformations that are not included in the **Tools > Filter** menu as individual commands. It is harder to use than the commands described before, but it can produce very interesting effects on glyphs. The Custom Filter is based on a predefined mathematical operation known as “convolution”, where each pixel is reassigned a value based on the values of surrounding pixels.

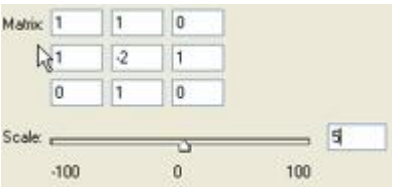
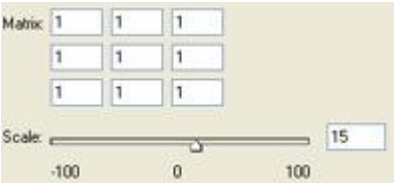
This command selection will bring up the Custom Filter dialog:



You will see ten text boxes in which to enter values. They are arranged as a square 3 by 3 matrix plus a scale factor. When you enter the **Matrix** values the **Scale** factor will automatically change, reflecting the matrix values sum. Use the **Scale** slider to change the factor; examine the preview to the left; then click on **OK** to apply the filter, or **Cancel** to dispose of the dialog.

To see the original glyph, press and hold down the **Ctrl** key. Surely you can undo this operation by using the **Undo** command.

Take some time to experiment with this filter. We will show you some examples here:

Original glyph	Filter values applied	Result glyph
		
		

As you can see the parameters in the last example allow you to perfect the glyph by removing unwanted black pixels outside and white holes inside the glyph's shape. This is a more general case of the ***Enhance*** (on page 292) operation.

Find Edges

This command allows you to get the contour of a glyph. As with other filters it can be applied to the selected part of a glyph, to the whole glyph while open in it is Glyph window, to all the glyphs in the font or to the selection in an Image window.

To apply this operation, select the **Find Edges** command in the **Tools > Filter** menu. As it has no customizable parameters, you will see the result immediately. For example:

Before

After

Use the **Undo** command to undo the operation.

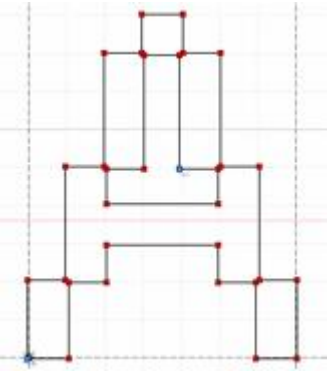
Outline Operations

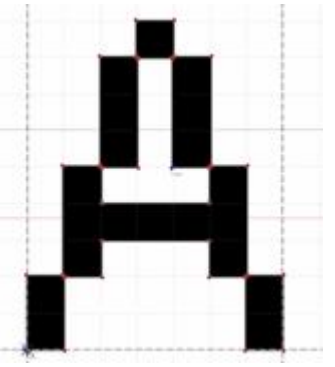
Operations from the **Tools** menu are applied not only to the glyph images but also to the outline layers if not empty. Operations from the **Tools > Adjust** and **Tools > Filter** menus do not affect outlines.

Additionally there are five special operations in the **Tools > Outline** menu for managing the glyph outline layer. Use them to manually create, delete, fill or rasterize glyph outlines.

To create an outline in the glyph outline layer, select the **Trace Bitmap** command in the **Tools > Outline** menu. BitFonter will autotrace the glyph image accordingly to the options set in the Trace Options dialog described later in the section and place the contours on the outline layer.

To create an outline with the simple pixel-based algorithm, use the **Trace Pixels** command in the **Tools > Outline** menu. BitFonter will trace the glyph image the same way that it does when exporting to an outline pixel font:



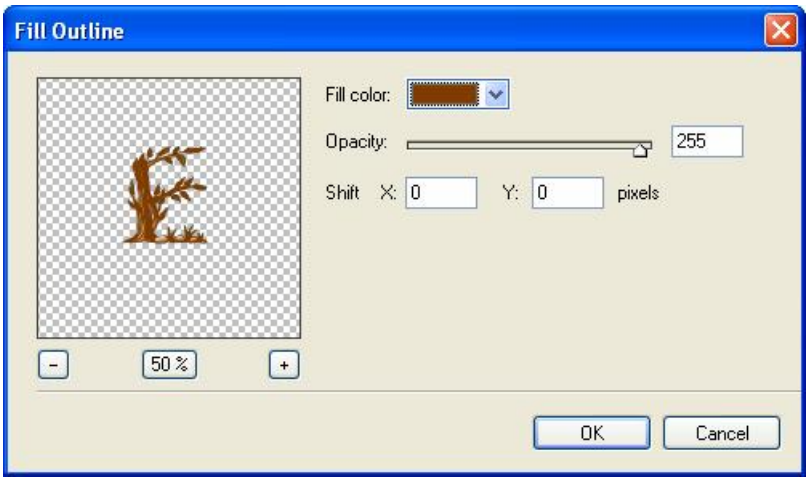


This command can be applied to black & white fonts only.

To clear the contents of the outline layer, select the **Clear** command in the **Tools > Outline** menu. The glyph contours will be removed.

To replace the bitmap contents of the glyph, select the **Rasterize** command in the **Tools > Outline** menu. The glyph contours will be rasterized and the new glyph image will be created. Note that color glyphs will be replaced with black & white.

To fill the glyph's outline with color pixels, select the **Fill** command in the **Tools > Outline** menu. The Fill Outline dialog appears:



Select **color** and **opacity** level, enter the values of horizontal and/or vertical **shift** if needed, and click on the **OK** button. The interior of the glyph's outline will be filled with the selected color with antialiasing. This operation preserves the original pixels of the glyph image that lay outside of the contours.

All the above outline operations can be applied to the whole glyph in the Glyph window or to several glyphs when they are selected in the Font window.

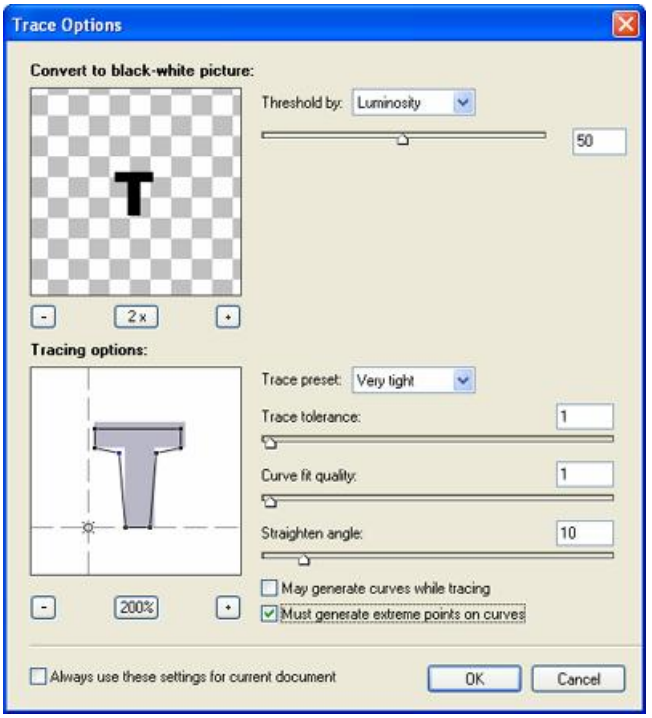
➘ Note: The content of the outline layer can be saved to the disk only in BitFonter Binary Font (.bfb) format, or it can be exported to outline-editing application.

There are two special commands in the **Edit** menu as well. If copied with the common **Copy** command glyphs have outlines in their outline layer, the glyph outlines can be pasted without bitmaps. Use the **Paste Outline** and **Append Outline** commands to paste only outlines when in the Font window.

Trace Options

Autotracing is used to create glyph outlines through the use of the **Trace Bitmap** command or when exporting glyphs to an outline font editing application. Some drag-and-drop operations also need the autotracer, for example, when glyphs are converted to outlines when dragged from BitFonter's Font window to an Adobe Illustrator image.

The Trace Options dialog is used to customize the autotracing feature of BitFonter. To open it when in the Glyph window, select the **Tools > Outline > Trace Bitmap** command while holding down the **Ctrl** key:



There are two separate parts of options in the dialog: **Black & White Conversion Options** and **Tracing Options**.

If a grayscale or color font will be traced, you must define the parameters of conversion of the glyphs to black & white in the upper part of the dialog. Choose the **Threshold by** conversion method in the dropdown list. Use the slider to adjust the threshold level if needed. (See the ***Threshold*** (on page 282) operation discussed earlier.) If a black & white font is autotraced, you will see the short version of the dialog without Black & White Conversion Options.

After you define the conversion method you can define the following tracing options:

Easy Trace Options

In the **Trace Preset** dropdown list you can quickly select common predefined options, changing from *Very tight* to *Very loose*. When you select one of these easy options in the dropdown list BitFonter will automatically adjust all the tracing parameters. The tighter the option you choose the more accurate the tracing will be. In other words, the outline will be closer to the original bitmap image. This is the first law of autotracing. The second law is that the tighter the option you choose the more nodes you will get on the outline. More nodes mean more time and larger font files.

Usually, the *Normal* option will be the best. If you find that the *Normal* autotracing option does not work for you, you can try the other tracing options listed in the **Trace Preset** dropdown list:

Glyph pixel size		
Small	**Medium**	**Large**
Tight	Normal /Tight	Normal
Tight/Normal	Normal	Loose
Normal	Normal	Loose

Advanced Trace Options

You can customize the trace parameters with the more detailed options below the **Trace Preset** dropdown list:

Trace tolerance	Allows you to change the distance between the generated outline and the edge of the original bitmap
Curve fit quality	Allows you to change the accuracy of curve fitting in the generated outline
Straighten angle	Defines the angle between two lines less than which the autotracer will replace several lines with one line
May generate curves	This option (active by default) allows the autotracer to generate curves
Must generate extreme points on curves	This option (active by default) forces the autotracer to insert nodes at the extreme points of curves

Every time you change a parameter the preview to the left also changes allowing you to check your results. Use the zoom buttons of the preview if needed.

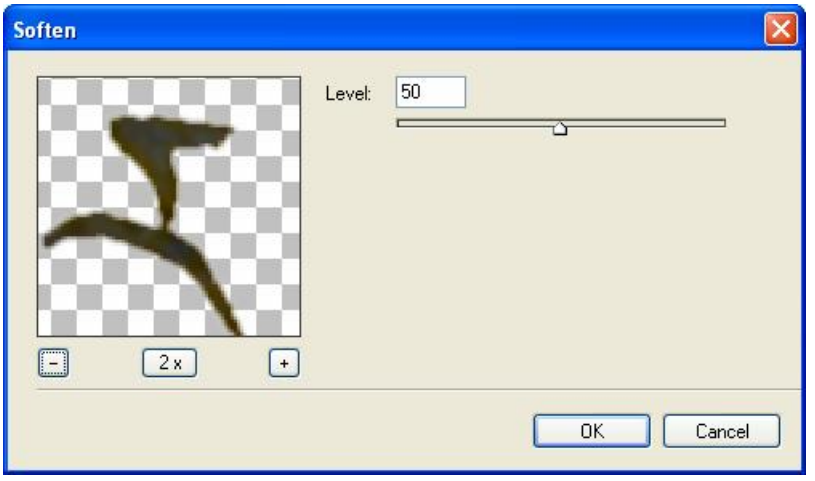
You can make the tracing options permanent for the current font by setting the **Always use these settings for current document** check box. In this case BitFonter will remember your choice and will not produce this dialog until you press the **Ctrl** key on the keyboard while choosing the **Tools > Outline > Trace Bitmap** command.

➘ Note: The trace options that you define in this dialog are used also when you select the Export > Outline Font feature described earlier in the ***Exporting Outlines*** (on page 214) section.

Example of Operations Usage

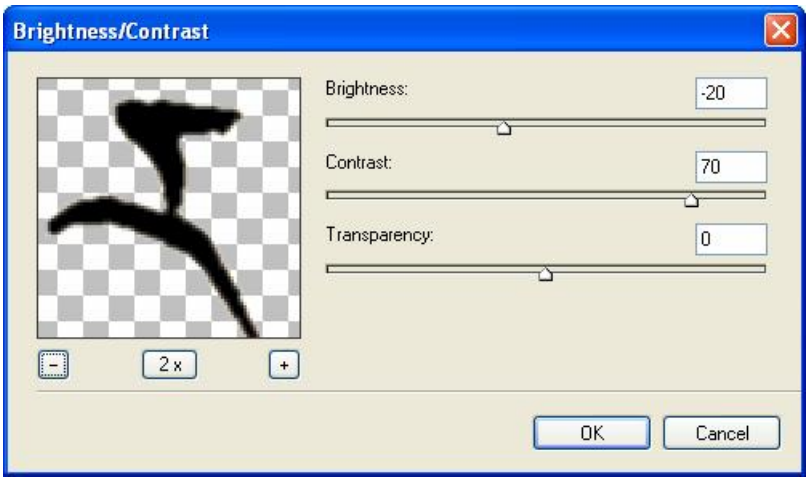
Let's take the font produced from the example image in the ***Creating a Font from an Image*** (on page 165) section. We have glyphs like this one:

To improve this and other glyphs, we will apply some filters. Let's make the glyph softer with the **Soften** filter. Select the glyphs in the font chart and choose the **Soften** command in the **Tools > Filter** menu. You will see the Soften dialog:



Use the slider to adjust the softness level and click on the **OK** button. The glyph's edges became better.

Now we will make the glyph darker and higher contrast with the Brightness/Contrast filter. Select this command in the **Tools > Adjust** menu and you will see the corresponding dialog:



Use the **Brightness** slider to make the glyph darker and the **Contrast** slider to make it higher contrast. Click on the **OK** button when finished and look at the difference between the source glyph and the result:

Original glyph *Result glyph*

Editing Metrics

The tools in BitFonter that you can use to edit metrics data are quite diverse. You can edit metrics:

- with the Edit Metrics tool in the Glyph window;
- in the glyph Properties panel;
- with the Generate Metrics, Adjust Width and Adjust Sidebearings operations;
- and in the Metrics window.

You can use all these methods or any of them that you feel more suitable for you.

Editing Metrics in the Glyph Window

This method was discussed already in the ***Edit Metrics Tool*** (on page 267) section of the ***Editing Glyphs*** (on page 225) chapter. Here is the short story.

The Edit Metrics tool is used for glyph and font metrics editing. Glyph metrics include the left and right margins of the glyph. They are horizontal metrics. Font metrics are the baseline, font ascent, font descent, Caps height and x height. These are vertical metrics.

To edit glyph metrics with the Edit Metrics tool, click on the mouse cursor on one of the lines and drag the line to its new place.

Editing the baseline affects only the open glyph; the glyph is shifted vertically.

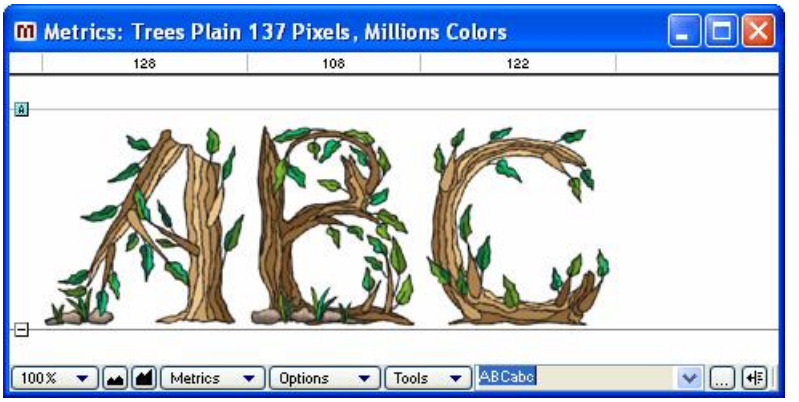
Dragging font metrics lines except the baseline always affects *font* metrics i.e. all the glyphs in the font.

Holding down the **Shift** key while the cursor is positioned over a metrics line (which changes the cursor to or) and clicking on the mouse will move the line to the edge of the glyph shape. This works with both glyph and font metrics.

Metrics Window

BitFonter has a special window where you can edit the metrics and kerning information. It is called the Metrics window.

To open the Metrics window select the **Metrics Window** command in the **Window** menu. The Metrics window will appear:



The Metrics window consists of several parts:

1. A Metrics Tools toolbar with four buttons that allow you to select one of the editing modes:

   By default this toolbar is vertically aligned and docked to the left side of the BitFonter main window. You can drag it anywhere or dock to any side.

2. A local command area that is used to select a mode for the Metrics window and a string for metrics or kerning editing:

3. The editing area where the edited string with controls appears.

4. The Metrics Ruler — a narrow bar located above the editing area:

Editing Modes

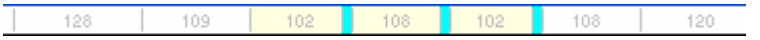
The Metrics window may work in four different modes:

Text mode	Is used to enter and edit text in the main editing area. Works very similarly to any standard text editor such as Notepad
Preview mode	This mode is used to preview text with kerning applied and check it at different sizes. Also the position and width of the underline line can be previewed in this mode
Metrics mode	This mode is used to adjust the metrics of individual glyphs. Kerning is not visible in the Metrics mode
Kerning mode	In this mode you can edit pair kerning.

Other things that appear in the Metrics window are the: Ruler and Table.

Metrics Ruler

The Metrics Ruler is a narrow bar located above the editing area:



Its purpose is very simple: to provide an overview of metrics and kerning data for the current line of text in the editing area.

The Metrics Ruler shows the advance width of the glyphs (in the middle of the glyph cell) and kerning. Kerning data appears on a light-blue background if kerning is negative (as in the AV pair) and on a yellow background when kerning is positive.

Of course, kerning information appears on the ruler only when the Metrics window is in kerning, text or preview mode.

You can control the appearance of the ruler using the **Ruler** command in the **Options** menu:

Metrics Table

Use this button in the bottom control area of the Metrics window to open the Table:

Name	Width	Left	Right
A	743	-1	0
B	660	31	64
C	746	48	38
D	813	29	42
E	690	46	29
F	598	32	-12
G	818	48	36
H	853	35	37
I	371	20	40
J	332	-77	16
K	712	56	-10

This table contains metrics or kerning information for all the glyphs in the font.

When the Metrics window is **in metrics mode**, every row in the table contains the name of the glyph, the glyph advance width and left and right sidebearings. If you right-click on the table header you will see additional fields to show in the table:

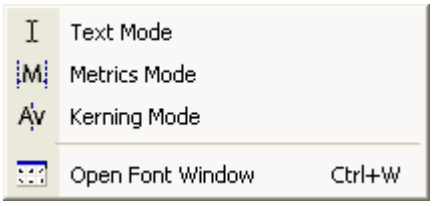
ScWidth	Scaled glyph advance width — the distance between the left and the right margins of the glyph in Units Per eM=1000
Bottom	Bottom sidebearing — vertical offset of the glyph image from the baseline
BdWt	Body width — the width of the glyph image in pixels. This field is not editable
BdHt	Body height — the height of the glyph image in pixels. This field is not editable.

In kerning mode every row contains the names of the first and second glyphs in the pair and a kerning value.

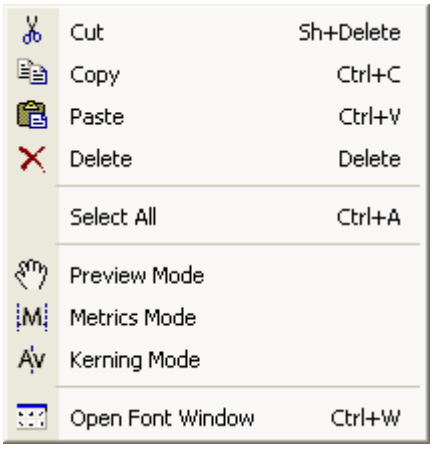
Context Menu

As in all other windows of BitFonter, if you right-click on the editing area, you will see a context menu that contains commands that are related to the current mode of the Metrics window.

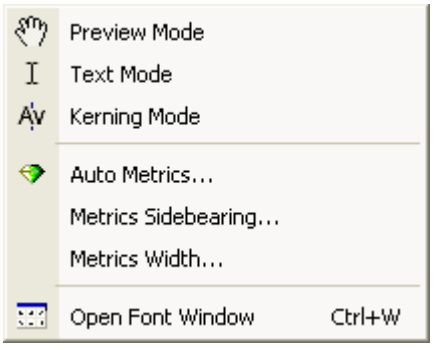
Every Metrics window mode has its own context menu:



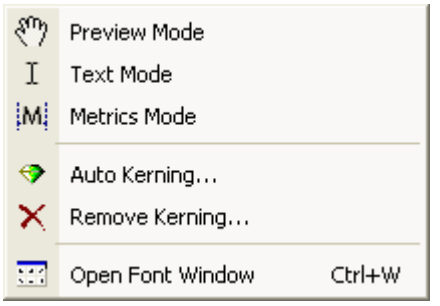
Preview mode context menu



Text mode context menu



Metrics mode context menu

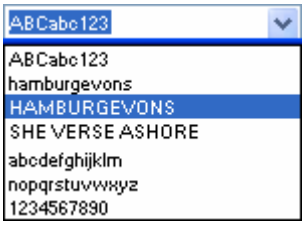


Kerning mode context menu

Selecting a String for Previewing or Editing

To prepare text for editing you have the following options:

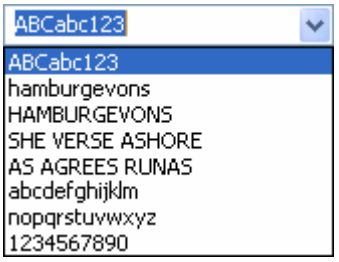
1. Select one of the predefined sample strings in the sample text dropdown list:



2. Enter the Text mode and type sample text directly in the Editing area.
3. Append glyphs to the Editing area by dragging them from other windows.

Selecting a Predefined Sample String

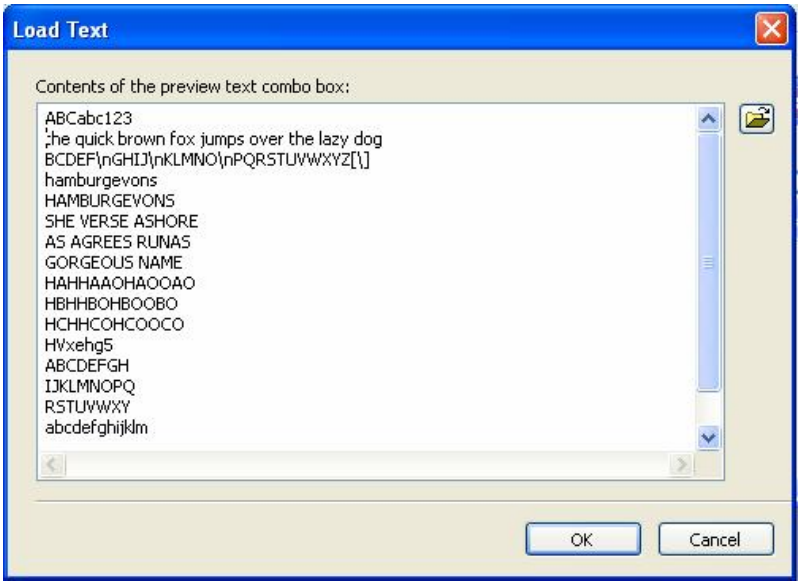
Use the sample text dropdown list to select the string for editing:



You can also use the **Ctrl+PgUp** and **Ctrl+PgDn** keyboard shortcuts to navigate the list of sample strings up and down.

Customizing the Sample String List

If you click on the button ... to the right of the sample list control, you will see the following dialog box:



As you can see, there is a big multiline editing field that contains all the strings in the sample list. Change it as you want or click on the button to fill it from a text file.

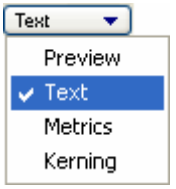
Type **\n** to force a line break in the sample text.

Enter some text, close the dialog box and then use the sample string scroll buttons or the **Ctrl+PgUp** and **Ctrl+PgDn** keyboard shortcuts to see how it works.

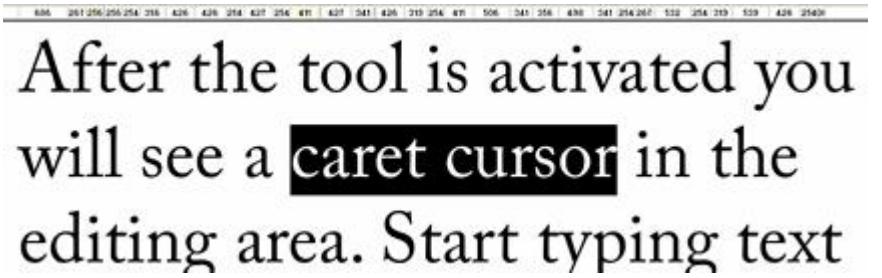
Entering Text in Text Mode

You may edit text in the editing area similarly to how you do it in any text editor. Activate the Text tool on the Metrics Tools toolbar:

You can also select Text in the **Mode Selection** menu in the local command area docked to the bottom:



After the tool is activated you will see a caret cursor in the editing area. Start typing text. You may also drag-select text and use the **Edit > Copy** and **Edit > Paste** commands to move blocks of text inside the Metrics window or from external applications.



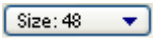
The copy-paste feature of the text tool is compatible with Unicode so if you paste some Unicode text, it will appear unchanged (if characters of that text are present in this font).

Using Drag-Drop

The easiest way to fill a sample string is **using the drag-drop method.** You can simply drag any glyph from the Font window and drop it in the Metrics window and it will be inserted in a position highlighted by the caret.

Selecting Preview Size

In the local command area you can see a **Size** menu:



and two buttons to the right of it:

Use these menu and buttons to decrease or increase the size of the sample string.

If the sample text becomes too large to fit in the window, a vertical scroll bar will appear allowing you to view all the editing areas of the Metrics window.

You can change the sample string behavior: switch on the **Auto Line Feed** option in the **Options** menu at the bottom of the Metrics window to not let the string expand outside the visible area of the window:

Previewing Metrics Lines

Some commands in the **View > Show Layers** menu work when the Metrics window is active:

	Glyph metrics	Glyph left and right sidebearings are visible in the current line
	Baseline	Font baseline is visible
	Font metrics	Font vertical metrics are visible in the current line.

Activating and Browsing Glyphs

Click on any glyph in the Editing area and it will be selected for further editing. In Metrics mode you will see the right and left handles that allow you to change the sidebearings and in Kerning mode you will see a pair handle that highlights a position between the first and second glyphs in the pair.

After you activate a glyph you can browse glyphs in the string. Use the "previous glyph" and "next glyph" shortcuts. By default they are **Ctrl + [** and **Ctrl +]** respectively.

In Metrics and Kerning modes you can change a glyph in the string by pressing the related key on the keyboards or by quickly entering its name.

Editing Metrics

This section discusses horizontal glyph metrics (the advance width and the sidebearings, jointly referred to as just *metrics*). In BitFonter you can modify this information either manually or automatically.

To modify glyph metrics, switch the Metrics window to the Metrics mode: click on the **Metrics Mode** button in the Metrics Tools toolbar or select the **Metrics** command in mode-selection menu on the bottom command area:

You can also right-click on the editing area and select the Metrics Mode command in the context menu.

Make sure the glyph metrics are switched on in the **View > Show Layers** menu.

Manual Metrics Editing

To modify a glyph's metrics you can use several methods:

1. Drag the sidebearings lines.
2. Drag the glyph within the editing area.
3. Edit the values in the Table.

To drag the sidebearings lines, position the mouse cursor on the line, hold down the left mouse button and drag the mouse. Release the left mouse button when you are done.

To drag a glyph within the editing area, position the mouse cursor on the glyph image; hold down the left mouse button and drag the mouse to position the glyph inside its advance width. Press and hold down the right mouse button while dragging the mouse to modify the glyph advance width.

You can also **modify the vertical position of the glyph** relative to its baseline. Hold down the **SHIFT** key on the keyboard while dragging the glyph. Make sure the baseline is switched on in the **View > Show Layers** menu.

Using the Keyboard

When the glyph is active you can use the keyboard to adjust the metrics:

Left and right arrow keys	Moves to the previous and next glyph in the sample string
Ctrl+left and right arrow keys	Moves the right sidebearing. This changes the right sidebearing and the advance width. Hold down the **SHIFT** key to move by 5 pixels at each key click on
Any character or digit	Selects the glyph you have typed as the current glyph for editing. You can also enter the glyph name if you want to access glyphs that are not assigned to any key combination
Ctrl+] and Ctrl+[	Moves to the next and previous glyph in the sample string.

Using the Metrics Table

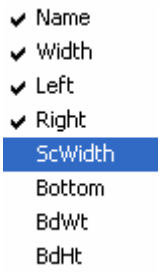
Use this button in the bottom control area of the Metrics window to open the Table:

Name	Width	Left	Right
A	743	-1	0
B	660	31	64
C	746	48	38
D	813	29	42
E	690	46	29
F	598	32	-12
G	818	48	36
H	853	35	37
I	371	20	40
J	332	-77	16
K	712	56	-10

The table in the Metrics editing mode may contain 8 columns:

Name	The name of the glyph. It is not editable here
Width	Glyph advance width — the distance between the left and the right margins of the glyph in pixels
Left	Left sidebearing — the horizontal offset of the glyph image from the glyph's left margin in pixels
Right	Right sidebearing — horizontal offset of the right margin from the glyph image in pixels(=Width — Left — BdWt)
ScWidth	Scaled glyph advance width — the distance between the left and the right margins of the glyph in Units Per eM=1000
Bottom	Bottom sidebearing — vertical offset of the glyph image from the baseline
BdWt	Body width — the width of the glyph image in pixels. This field is not editable
BdHt	Body height — the height of the glyph image in pixels. This field is not editable.

You can customize the metrics table view. To hide those columns that you do not need, right-click on the table header and select the item in the dropdown list:



Repeat this operation to make the column reappear in the table.

To change the width of the column, just drag its right boundary.

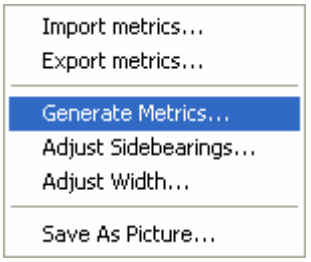
To edit a value in the table just click on it with the mouse cursor and enter the new value. You have the opportunity to select several rows by **Shift**-clicking on them and changing all of them at the same time.

➘ Note: You need not save any changes when in the Metrics window, just close it when finished editing and save the font. Note that kerning data can be only stored in BDF fonts and photofonts.

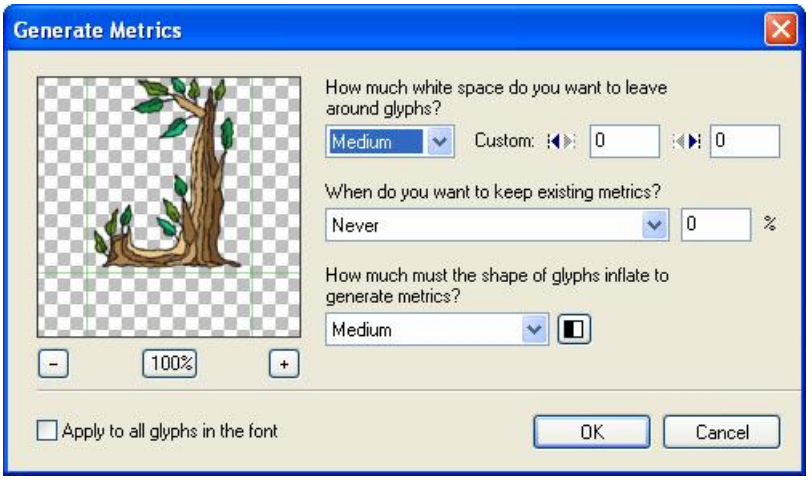
Automatic Metrics Generation

BitFonter can automatically define a glyph's metrics using a special algorithm. This algorithm usually produces good results but we recommend further manual editing for the best results.

To generate glyph metrics automatically, select the **Metrics > Generate** command in the **Glyph** menu or the **Generate Metrics** command in the local **Tools** menu at the bottom of the Metrics window:



The Generate Metrics dialog box appears:



Three dropdown lists are used to define the main data for the metrics generation algorithm. These parameters produce different results for different fonts. We recommend that you experiment with them and choose the appropriate result. Adjust the metrics manually later.

The button opens the **Black & white conversion defaults** page of the Font Info dialog box described in the ***Black/White Conversion*** (on page 113) section. Threshold is used here to help BitFonter find the glyph image boundaries if the font is not black & white. Finding the image boundaries is important for the algorithm. Also see the description of this parameter in the ***Threshold*** (on page 282) section in the ***Editing Glyphs*** (on page 225) chapter.

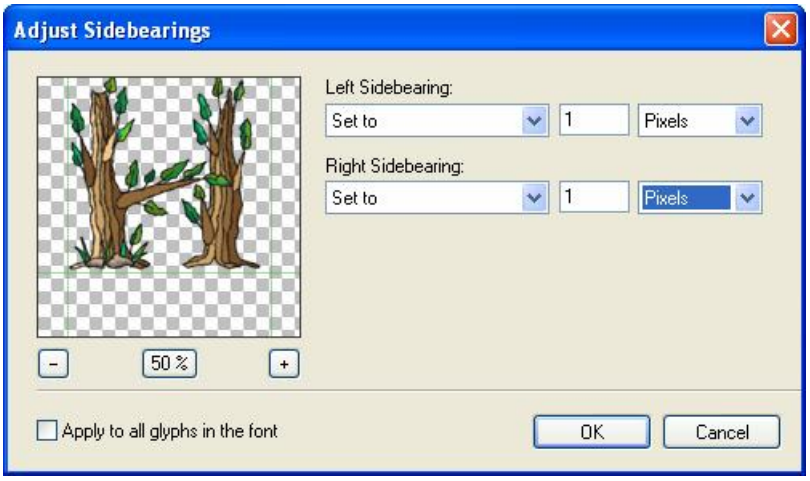
Switch on the **Apply to all glyphs in the font** option if you want new metrics be generated for all glyphs and click on **OK**. Otherwise the operation will be applied to the current glyph selected in the Metrics window only.

You can use the **Undo** command to undo this operation.

Adjusting Glyph Sidebearings

The Adjust Sidebearings operation is a useful addition to the Edit Metrics tool and the Generate Metrics operation. It allows you to automatically change the left and right margins of any number of font glyphs. As the other operations, the Adjust Sidebearings operation is applicable to the glyph in the Glyph window or to the selected glyphs when in the Font window.

When you select the **Sidebearings** command the **Glyph > Metrics** menu or the **Adjust Sidebearings** command in the **Tools** menu at the bottom of the Metrics window, the Adjust Sidebearings dialog appears:



Set the following parameters of the operation:

Left sidebearing	Use this option if you want to change the position of the left margin. You can set the absolute or relative values in pixels or percents to the current value
Right sidebearing	Use this option if you want to change the position of the right margin. You can set the absolute or relative values in pixels or percents to the current value.

You can view the preview of the glyph with its margins in the preview field of the dialog. To see the original margins position, press and hold down the **Ctrl** key.

Switch on the **Apply to all glyphs in the font** option if you want the operation be applied to all glyphs. Click on the **OK** button to apply the changes when finished, or **Cancel** to abort the operation.

You can use the **Undo** command to undo this operation.

Adjusting Glyph Advance Width

The Adjust Width operation is a useful addition to the Adjust Sidebearings operation. It allows you to automatically change the advance width of any number of font glyphs. As the other operations, the Adjust Width operation is applicable to the glyph in the Glyph window or to the selected glyphs when in the Font window.

When you select the **Width** command the **Glyph > Metrics** menu or the **Adjust Width** command in the **Tools** menu at the bottom of the Metrics window, the Adjust Width dialog appears:

Set the following parameters of the operation:

Glyph Width	Use this option if you want to increase or decrease the glyph advance width (the distance between its left and right margins) or set it to some particular value
Alignment	Use this option to align the glyph to the left or right margin, or to center it within the glyph advance width.

You can view the preview of the glyph with its margins in the preview field of the dialog. To see the original margins position, press and hold down the **Ctrl** key.

Switch on the **Apply to all glyphs in the font** option if you want the operation be applied to all glyphs. Click on the **OK** button to apply the changes when finished, or **Cancel** to abort the operation.

You can use the **Undo** command to undo this operation.

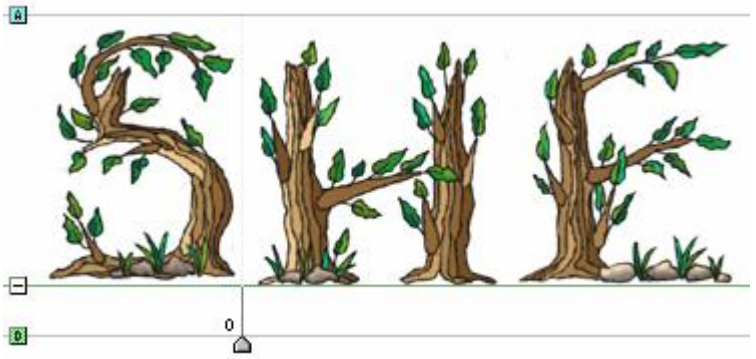
Editing Kerning

This section discusses font kerning. In BitFonter you can modify kerning either manually or automatically.

To modify kerning info, switch the Metrics window to the Kerning mode: click on the **Kerning Mode** button in the Metrics Tools toolbar or select the **Kerning** command in mode-selection menu on the bottom command area:

You can also right-click on the editing area and select the **Kerning Mode** command in the context menu.

To make the kerning editing controls visible you must select the second glyph of the pair that you want to edit. Position the mouse cursor on the right glyph of the pair in the editing string and click the left mouse button.



You will see the kerning line and handle appear in the editing area. Drag it to the left closer to the first glyph in the pair. There is now a blue area in the metrics ruler at the top part of the editing area. This means that kerning exists for that pair of glyphs and it is negative. The positive kerning values are marked with yellow.

You also can edit kerning with **Arrow** keys while holding down the **Ctrl** key. The **Shift** key allows you to change kerning value by 5 pixels at a click. Clicking on **Arrows** without **Ctrl** will change the current glyph.

➘ Note: The kerning data can be saved only in BitFonter documents, BDF and PhF fonts. Other bitmap font formats do not support kerning.

Using the Metrics Table

If the metrics table is visible (if not, use the button on the bottom panel to show it) in Kerning mode it will have three columns:

1st	2nd	Value
A	Q	-15
A	T	-79
A	U	-28
A	V	-69
A	W	-54
A	X	-2
A	Y	-87
A	a	4
A	b	-3
A	c	-12
A	d	-12
A	e	-12
A	f	-8
A	g	-12
A	h	-3
A	i	-3
A	j	-4

The first two columns contain the names of the first and second glyphs of the pair. The third column contains the kerning value.

You can easily change the kerning for a pair — click on the number in the right column and enter the new value. Click on **Enter** to accept or **Esc** to cancel.

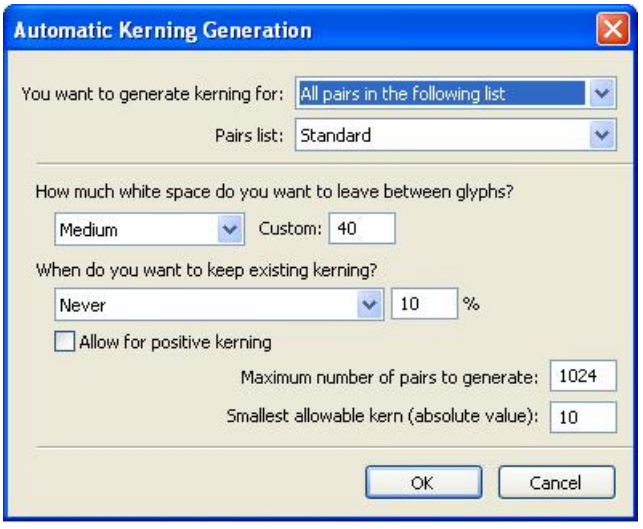
To delete a kerning pair, select it in the Metrics table and press the **Delete** key on the keyboard. The pair will be removed. Of course, you can use the **Edit > Undo** command to get your pair back.

Automatic Kerning Generation

The easiest way to apply kerning to a font is to use BitFonter's autokerning algorithm. This algorithm analyzes the shape of the glyphs in the given pairs and automatically kerns them. You can control the pairs list that the autokerning algorithm processes as well as other parameters.

To define kerning automatically select the **Auto Kerning** command in the **Glyph > Metrics** menu or in the context menu of the Metrics window.

The Automatic Kerning Generation dialog box appears:



This dialog box consists of two areas: the **Area of application** and **Parameters**.

In the first area you select the pairs for which the algorithm will compute kerning values. You can choose between **Current pair only** (available if one of the pairs is selected in the editing area), **All Pairs in the current string,** or **All Pairs in the following list**.

BitFonter allows you to generate kerning for all the pairs located in a special list file. The list files are stored in the [Shared default data]\Kerning folder (typically, C:\Program Files\Common Files\FontLab\Kerning). You can create your own kerning pair files or use one of the files placed there at the time of BitFonter's installation.

The **Parameters** area lets you customize the autokerning algorithm. The most used option is: **How much white space do you want to leave between glyphs?** This controls how close the glyphs will be moved together while computing kerning in the pair.

The **Allow for positive kerning** check box lets the autokerning algorithm produce positive kerning in pairs. Positive kerning moves glyphs apart from each other. Positive kerning is usually not recommended but there may be occasional circumstances where it is needed.

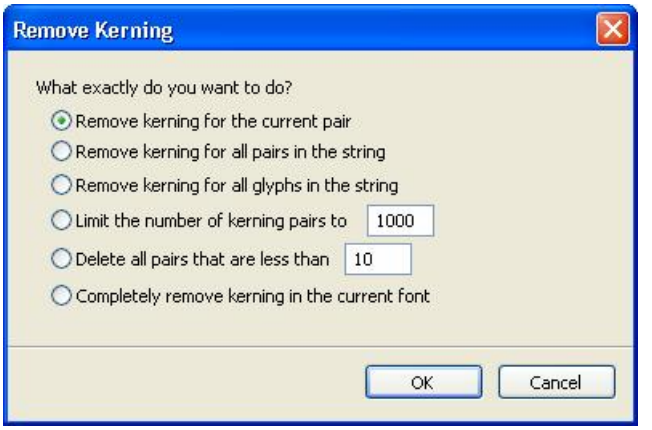
If you want to save the existing kerning the dropdown list lets you control the disposition of the existing (imported or manually created) kerning pairs. You can replace existing pairs by automatically generating new ones, keep them unchanged, or select the condition mode.

The **Maximum number of generated pairs** and **Smallest allowable kern** options control the possible number of automatically created pairs and the minimal normal (negative or positive) kerning value.

Resetting Kerning

To remove the kerning information for some glyphs or for the entire font you must use the Remove Kerning feature. To open the Remove Kerning dialog box select the **Remove Kerning** command in the **Glyph > Metrics** menu or in the context menu of the Metrics window.

The Remove Kerning dialog box appears:


Remove Kerning
What exactly do you want to do?
Remove kerning for the current pair
Remove kerning for all pairs in the string
Remove kerning for all glyphs in the string
Limit the number of kerning pairs to
1000
Delete all pairs that are less than
10
Completely remove kerning in the current font
OK
Cancel

This dialog box includes options that control kerning removal.

Available options are:

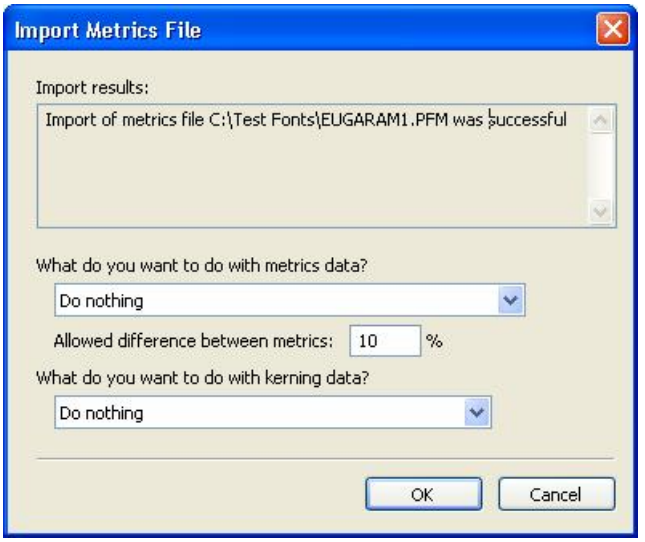
Remove kerning for the current pair	Removes kerning for the selected pair only
Remove kerning for all pairs in the string	Removes kerning in all pairs that exist in the current string
Remove kerning for all glyphs in the string	Removes kerning in all pairs that include glyphs in the current string
Limit number of kerning pairs to...	Kerns only the given number of pairs with the largest absolute kerning value
Delete all pairs that are less than...	Removes all pairs that have a kerning value less than the given value. The absolute value of kerning is compared
Completely remove kerning in the current font	Removes all the kerning pairs available in the font. Before doing this, we recommend you to save the current metrics and kerning data in the metrics file.

Opening Metrics Files

BitFonter allows you to import metrics and/or kerning information into the current font. Using this feature, you can create metric and kerning information once and use it in several similar fonts.

To import a metrics file into BitFonter select the **Metrics** command in the **File > Import** menu or the **Import Metrics** command in the **Tools** menu at the bottom of the Metrics window. You will see the standard Open File dialog box. Select the metrics file that you want to import (in PFM or AFM format) and click on the **Open** button.

The Import Metrics dialog box appears:



The topmost control contains a legend describing the metrics file that you are importing and its compatibility with the current font.

The options in the **Parameters** area let you select various metrics importing options:

What do you want to do with the metrics data:

Do nothing	Do not import metrics data from this file
Replace all metrics in the current font	Import all metrics data (glyphs' advance widths and sidebearings) and replace the metrics data in the current font. We recommend that you use this option only if your font is very similar to the metrics file that you are importing
Replace all metrics that are close to current	Replace only those metrics records that are similar to the imported metrics. The **Allowed difference between metrics** option controls the allowed difference
Replace metrics that are thicker than in the current font **Replace metrics that are wider than in the source font**	These options are obvious.

What do you want to do with the kerning data:

Do nothing	Do not import kerning data from the metrics file
Completely replace kerning data in the current font	Remove all existing kerning pairs and replace them with pairs imported from the metrics file
Add imported kerning data to the current font	Leave the existing kerning pairs unchanged but add new kerning pairs from the metrics file
Add new kerning pairs but autokern them	Import information about the glyphs that form each kerning pair in the metrics file and apply an autokerning algorithm to these pairs.

Select the appropriate options and click on **OK** to proceed or click on **Cancel** to abort the operation.

Saving Metrics Files

Sometimes you may need to export all current metrics and kerning in a separate metrics file. You can always do so by using the Metrics window. Select the **Metrics** command in the **File > Export** menu or the **Export Metrics** command in the **Tools** menu at the bottom of the Metrics window. The standard Save File dialog box appears.

Select the destination format (AFM or PFM), and the destination directory. Enter the file name and click on the **Save** button to save the metrics file.

Index

A

AAT, 22
Adobe, 22
advance width, 61, 92, 100, 117, 224, 231, 319
AICB option, 169
Append, 300
appending
  chracters, 134
Arabic, 122
ascender, 53, 98, 231
ASCII, 22, 117
AsiaFont Studio, 213
ATSUI, 22
autosplitting, 180
  customizing, 184
average width, 97

B

background
  removing, 178
baseline, 193, 231
  aligning, 180
  editing, 180, 308
  moving, 180
batch conversion, 74
BDF, 107, 206
bit map, 26
BitFonter, 9
  features, 9
  interface options, 47
  options, 44
  user interface, 44
BitFonter Options, 82, 84, 86, 188, 197, 239, 269
BitFonter User Interface, 10, 213
bitmap, 26
  pasting, 269
  pasting options, 53
bitmap fonts, 26
Bitmap Pasting, 251
Black/White Conversion, 324
bounding box, 53, 180
Brightness/Contrast, 280
brush
  angle, 258
  diameter, 258
  hardness, 258
  roundness, 258
Brush tool, 258
Brushes Panel, 258

C

canvas size, 176
caps height, 97
Cell tool, 185
cells
  defining, 180
  exporting, 196
  mask, 185
  merging, 180
  splitting, 180
Changing Font Color Mode, 164, 174
Changing Font Size and/or Resolution, 164, 273
characters
  appending, 134
  copying, 133
  creating, 131
  deleting, 135
  moving, 132

pasting, 133
reencoding, 136
renaming, 136
searching for, 130
selecting, 128
transforming, 135
Clipboard, 66, 166, 170, 171, 251, 269
codepage, 22, 103, 104, 105, 119, 122, 140, 142
custom, 26
codepage definition, 122
Codepages, 104
color
selecting, 253
color depth, 26
color fonts, 26, 71
saving, 210
color mode, 150, 162, 174
Color Palette, 252, 255
color Picker, 253
color table, 150, 162, 174
Colorize, 285
Context Menu, 116, 242, 313
copyright
information, 94
Creating a Font from an Image, 305
Creating a New Font, 197
Creating and Saving Fonts, 73, 83, 84
Custom Filter, 292, 297
Cyrillic, 121, 134

D

default character, 103
Define Cell Tool, 185
demo version, 12
descender, 98, 231
dithering, 150, 174

E

EBSC table, 56
Edit Metrics Tool, 308
Editing Family Properties, 75
Editing Font Properties, 75, 88, 89, 148
Editing Fonts, 75, 273
Editing Glyph Properties, 136
Editing Glyphs, 87, 113, 126, 135, 136, 148, 151, 152, 173, 175, 308, 324
Editing Metrics, 126, 158
Editing Metrics in the Glyph Window, 232
Editing Style Properties, 75
Ellipse tool, 263
embedded bitmap, 56, 84
embedded fonts, 71, 84
Embedding Mark, 145
Emboss, 289
Encoding, 22, 102, 120
encoding vector, 119, 120, 140
Enhance, 292, 298
EPS, 166, 169
Eraser tool, 261
export
options, 56
Exporting Outlines, 233, 304
Extended Latin, 121
extra properties, 111
Eye Dropper tool, 253

F

family
info, 88
name, 103
Fill tool, 261
Find Edges, 299
Find Glyph, 28, 130
Flip Horizontal, 272
Flip Vertical, 272
folders and paths, 19
FOND ID, 103
'FOND' resource, 22
font
adding to project, 145
copying, 74
creating, 73, 162, 165
creating from image, 165
deleting, 74
dimensions, 97
duplicating, 74, 147
exporting, 212
importing, 198
metrics, 238, 267, 308

pasting, 74
properties, 75, 90
reencoding, 140
removing, 74, 146
resolution, 148
reverting, 144
saving, 203
font color mode
changing, 150
Font Family, 27
creating, 73
properties, 88
reencoding, 142
font name, 92
Font Project, 65
creating, 67
opening, 69
properties, 76
saving, 81
Font Set, 65, 83
saving, 83
font size
changing, 148
font window, 114
background color, 117
cells, 117
customizing, 48, 127
editing mode, 119
footer, 115
using, 125
Font Window Options, 127, 226
FontLab, 26, 198, 213, 214
importing from, 201
FontLab template, 203
fonts
blending, 152
FreeHand, 169
full name, 92

G

Gaussian Blur, 291
glyph, 22, 28
advance width, 61, 92, 100, 117, 224, 231, 319
code, 117
drawing, 252
metrics, 267, 308
name, 117, 234
names, 196
properties, 116, 234, 242
sidebearings, 231, 234
structure, 229
transforming, 270
transparency, 256
Unicode, 117, 234
zooming, 127
glyph cell
caption, 117, 127
Glyph window
changing view, 236
colors, 50
context menu, 242
customizing, 50
Glyph Window Options, 232
glyphs
appending, 134
creating, 131
deleting, 135
searching for, 130
transforming, 135

H

Hand tool, 236
Hebrew, 121
HP Soft font, 103
Hue/Saturation, 281

I

Illustrator, 169
image
colors changing, 174
conversion, 61
creating, 171
editing, 173
exporting, 222, 224
importing, 168, 170
preparing, 166
size, 168, 176

splitting, 180
Image Window, 180
import
options, 55
Importing a Font, 233
Info Panel, 240, 253
Invert, 284
italic angle, 100

K

Kerning
editing, 328, 329
keyboard shortcuts, 185, 243
customization, 40
Knife tool, 185

L

leading, 97
Line tool, 261

M

Macintosh, 120
Magic Wand tool, 248
Make Bold, 278
Median Noise, 294
menu
customizing, 35
Toolbars, 66
View, 66
metrics, 97, 231
editing, 307, 320
metrics window, 309
Kerning mode, 328
Metrics mode, 319
Metrics Tools, 309
modes, 310
opening, 158
Text mode, 316
Metrics Window, 232
Mouse, 21
MS Charset, 103

N

name table, 119, 120, 140
names table, 136
NeXT Step, 122

O

Opaque mode, 250
OpenType, 22, 26, 67, 69, 73, 217
OpenType TT, 67, 69, 81
saving, 84
operation
adjust glyph metrics, 325, 327
Brightness/Contrast, 280
Colorize, 285
Custom Filter, 297
Emboss, 289
Enhance, 292
Find Edges, 299
Flip Horizontal, 272
Flip Vertical, 272
Gaussian Blur, 291
generate glyph metrics, 323
generate kerning, 330
Hue/Saturation, 281
Invert, 284
Make Bold, 278
Median Noise, 294
Reduce Noise, 293
Rotate, 275
Scale, 273
Sharpen, 295
Shift, 279
Slant, 277
Smart Blur, 290
Soften, 296
Threshold, 282
Operation menu, 135
Operations, 135, 148, 152, 251
Operations toolbar, 270
options, 44
OS/2, 122
outline, 26, 169
creating, 300
deleting, 300
filling with color, 300
rasterizing, 300

outline font editor, 62, 201, 213
Outline Font Editor, 213
outline fonts, 26
outline layer, 233, 238
Outline Layer, 200
Outline Operations, 133, 200, 216, 233, 242
outlines, 180, 199, 201, 214

P

Painter, 9
Palette Panel, 255
Palm, 67, 69, 81, 85, 203, 208, 210
Palm OS font
  saving, 85
PCL Character Set, 103
Pencil tool, 257
PhotoFont, 56, 67, 69, 73, 81, 86, 152, 203, 209, 210
  saving, 86, 209
Photoshop, 9
pixel outline font, 217
pixel size, 71, 97, 148, 162, 176
Pixels
  tracing, 300
PNG, 166, 222, 285
point size, 97, 162
Polygon tool, 262
PostScript, 22, 26, 120
project, 65, 199
  creating, 67
  info, 76
  properties, 76
  reverting, 80
  saving, 81
  saving a copy, 82
  verifying, 79
  version, 76
project window, 70
  contents, 71
  context menu, 72
  toolbar, 71

R

RAM, 12
Rectangle tool, 263
Reduce Noise, 293
Reencoding Glyphs Automatically, 102
Removing a Font from the Project, 74
resampling, 48, 148, 176
resolution, 53, 97, 162, 230
  changing, 148, 176
Rotate, 275

S

Sample font, 27
Saving a Font, 212
Saving a Font Project, 71
Saving Color Fonts, 84, 203
Saving in Windows Font Format, 83, 84
Scale, 273
scale bar, 190
scale factor, 190
scanner, 166
script, 103
Select Region tool, 247
Select tool, 246
Selecting, 265
selection
  copying, 249, 251
  deleting, 251
  moving, 249
  pasting, 251
  transforming, 251
Sharpen, 295, 296
Shift, 279
sidebearing, 61, 320, 325
Slant, 277
Smart Blur, 290
Soften, 295, 296
spacing, 92, 162
splitting
  editing, 185
splitting data
  saving and restoring, 189
Standard toolbar, 66
strikeout, 99
style
  properties, 89
subscript, 99
suitcase, 22

superscript, 99
supported codepages, 104, 105

T

Tablet Options, 258
template characters, 127
Text tool, 264
Threshold, 151, 175, 181, 282, 293, 303, 324
TIFF, 166, 222, 285
Tool Options panel, 244, 248, 250, 258, 268
toolbar
  Metrics Tools, 309, 319
  Operations, 270
  Standard, 66
  Tools, 185, 243
Tools toolbar, 243
Trace Options, 215
Tracing Options, 214, 302
transparency, 22, 238, 250, 284
Transparent mode, 250
TrueType, 22, 26, 108
  saving, 56, 81, 84
TWAIN, 166
Type 1, 26, 217
TypeTool, 26, 213, 219

U

underline, 98
Unicode, 115, 121, 131, 136, 196
  assigning, 136
Unicode range, 22, 119, 121

V

View Menu, 238
View mode
  selecting a string, 314

W

Window menu, 158
Windows, 122
Windows Font Set, 69, 81, 83
Working with an Image, 288

X

x height, 98, 231

Z

Zoom tool, 268
Zoom Tool, 237